

Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Naváděcí systém analyzující obrazový signál
Guidance System based on Image Signal Analysis

2019

Jan Durčák

Zadání bakalářské práce

Student:

Jan Durčák

Studijní program:

B2649 Elektrotechnika

Studijní obor:

2612R041 Řídicí a informační systémy

Téma:

Naváděcí systém analyzující obrazový signál
Guidance System based on Image Signal Analysis

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Rozbor problematiky zpracování obrazového signálu.
2. Návrh systému pro navádění s využitím obrazového signálu.
3. Realizace systému pro navádění s využitím obrazového signálu.
4. Měření a analýza navádění s vyhodnocením skutečných parametrů systému.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] SOJKA, Eduard. *Digitální zpracování a analýza obrazů*. Ostrava : VŠB - Technická univerzita Ostrava, 2000. ISBN 80-7078-746-5.
- [2] SOBOTKA, Zdeněk a Martin SOBOTKA. *Počítačová analýza a rozpoznávání obrazu*. Praha: Dům techniky Československé vědeckotechnické společnosti, 1990. ISBN 80-02-00739-5.
- [3] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. Praha: BEN-Technická literatura, 2008. ISBN 978-80-7300-233-6.
- [4] SMITH, Steven W. *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego, California: California Technical Pub., 1997. ISBN 978-0966017632.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeněk Macháček, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem svou bakalářskou práci na téma „Naváděcí systém analyzující obrazový signál“ vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 29. dubna 2019


podpis studenta

Poděkování

Rád bych zde poděkoval panu Ing. Zdeňku Macháčkovi, Ph.D. za odborně vedenou pomoc při konzultacích a za rady, které mi během vytváření bakalářské práce věnoval. Také bych chtěl poděkovat svému bratranci Jiřímu Čechovi za užitečné rady a poskytnutou výpočetní techniku.

Abstrakt

Práce se zabývá naváděcím systémem analyzujícím obrazový signál. Pojednává o systému, jež snímá kamerou jistou oblast. V této oblasti se může či nemusí nacházet referenční objekt. Pomocí kontinuálního obrazového signálu, snímaného kamerou, lze rozpoznat aktuální polohu objektu. Systém pozici dokáže vykreslit do obrazu a vypsát hodnoty objektu. Zjišťované hodnoty potřebné na navádění jsou vzdálenost a úhel vůči předem určenému cíli, rychlost snímání obrazu a přesnost rozpoznání referenčního objektu a jeho směrového bodu. Zjištěné hodnoty jsou vypsány do obrazu či do konzole a lze je použít pro samotné navádění.

Klíčová slova

Zpracování obrazu, navádění, lokalizace, referenční objekt

Abstract

This work concerns itself with a guiding system which analyses display signal. It deals with a system which captures a certain area with a camera. The reference object may or may not be in this area. By the means of the continuous display signal, captured by the camera, it can be recognized the current position of the object. The system can render the position in the image and list the object's values. The wanted values required for guidance are the distance and angle to the predetermined target, the frame rate of the system, and the accuracy of recognition of the reference object and its directional recognition. Detected values are listed in the image or console and can be used for guidance.

Key words

Display processing, guidance, localization, reference object

Obsah

Seznam použitých symbolů.....	- 8 -
Seznam použitých zkratek.....	- 9 -
Seznam obrázků	- 10 -
Seznam tabulek	- 11 -
Úvod.....	- 12 -
1 Rozbor analýzy obrazového signálu	- 13 -
1.1 Rozbor obrazu	- 13 -
1.2 Rozbor signálu.....	- 14 -
1.2.1 Digitalizace signálu	- 14 -
1.2.2 Barevné modely.....	- 15 -
1.3 Předzpracování obrazu	- 18 -
1.3.1 Jasové transformace	- 18 -
1.3.2 Geometrická transformace.....	- 20 -
1.3.3 Filtrace.....	- 20 -
1.4 Segmentace.....	- 21 -
1.5 Detekce hran.....	- 21 -
1.5.1 Hrana	- 21 -
1.5.2 První derivace.....	- 23 -
1.5.3 Druhá derivace	- 23 -
1.5.4 Srovnání se vzorem	- 23 -
2 Návrh systému.....	- 24 -
2.1 Softwarové řešení.....	- 24 -
2.2 Rozbor problematiky navádění	- 24 -
2.3 Referenční objekty	- 25 -
2.4 Umělá neuronová síť	- 27 -
2.5 Strojové učení.....	- 28 -
2.6 Návrh hardwarových komponent naváděcího systému	- 29 -
3 Realizace systému	- 30 -
3.1 Řešení hardwarových komponent	- 30 -
3.1.1 Jednočipový počítač Arduino	- 30 -
3.1.2 Jednodeskový počítač Raspberry Pi	- 30 -

3.1.3	USB kamera	- 31 -
3.1.4	Kamerový modul Raspberry Pi	- 31 -
3.1.5	Realizace testovacího přípravku	- 31 -
3.2	Implementace učícího procesu neuronové sítě.....	- 32 -
3.3	Implementace procesu naváděcího systému.....	- 34 -
3.4	Vizualizace originálního naváděcího systému analyzujícího obrazový signál	- 36 -
4	Měření a analýza navádění.....	- 39 -
4.1	Verifikace rychlosti snímání	- 39 -
4.2	Verifikace přesnosti snímání vzdáleností.....	- 39 -
4.3	Verifikace přesnosti snímání úhlů.....	- 40 -
4.4	Verifikace rozpoznání referenčních objektů	- 41 -
4.5	Analýza rozdílů použitých kamerových zařízení	- 41 -
5	Zhodnocení dosažených výsledků.....	- 42 -
	Závěr	- 43 -
	Použitá literatura	- 44 -
	Seznam příloh.....	xlvi
	Seznam příloh na DVD	xlvi

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
λ	m	Vlnová délka
F	Hz	Frekvence
C	$\text{m} \cdot \text{s}^{-1}$	Fázová rychlost
t	s	Čas
s	m	Vzdálenost
%		Procento
α	°	Úhel

Seznam použitých zkratek

Zkratka	Význam
RGB	Red Green Blue
CMY	Cyan Magenta Yellow
CMYK	Cyan Magenta Yellow Key
HSV	Hue Saturation Value
px	Pixel
fps	Frames per second
USB	Universal Serial Bus

Seznam obrázků

Obrázek 1.1: Rozdělení dle vlnové délky	- 13 -
Obrázek 1.2: Čtvercová a hexagonální vzorkovací mřížka	- 15 -
Obrázek 1.3: Aditivní míchání barev	- 16 -
Obrázek 1.4: Subtraktivní míchání barev	- 16 -
Obrázek 1.5: 3D RGB model	- 17 -
Obrázek 1.6: 3D CMY model	- 17 -
Obrázek 1.7: Globální transformace	- 19 -
Obrázek 1.8: Lokální transformace	- 19 -
Obrázek 1.9: Bodová transformace	- 20 -
Obrázek 1.10: Grafy různé intenzity detekované hrany	- 22 -
Obrázek 2.1: Vývojový diagram procesu navádění	- 25 -
Obrázek 2.2: Referenční objekt čtverec	- 26 -
Obrázek 2.3: Referenční objekt kruh	- 26 -
Obrázek 2.4: Umělý neuronový model	- 27 -
Obrázek 2.5: Propojení umělých neuronů v umělé neuronové síti	- 28 -
Obrázek 2.6: Návrh propojení hardwarových komponent	- 29 -
Obrázek 3.1: Schéma Raspberry Pi	- 31 -
Obrázek 3.2: Kvádrový stojan na tetovací přípravek	- 32 -
Obrázek 3.3: Fotografie objektů v prostředí	- 33 -
Obrázek 3.4: Fotografie objektů v prostředí se zvýrazněním objektů	- 33 -
Obrázek 3.5: Graf odchylky učicího krokovacího procesu	- 34 -
Obrázek 3.6: Vývojový diagram naváděcího systému analyzujícího obrazový signál	- 35 -
Obrázek 3.7: Rozeznané referenční objekty a jejich směrové body	- 36 -
Obrázek 3.8: Vzdálenosti vypsane do obrazu v px	- 36 -
Obrázek 3.9: Vzdálenosti vypsane do konzole v cm	- 37 -
Obrázek 3.10: Úhel natočení objektu vůči cíli	- 37 -
Obrázek 3.11: Směrové šipky ukazující směr k cíli	- 37 -
Obrázek 3.12: Objekt v cíli	- 38 -
Obrázek 4.1: Graf porovnání rychlosti snímání obou kamerových systémů v čase	- 39 -

Seznam tabulek

Tabulka 1.1: Rozdělení spektra elektromagnetických vln [7]	- 14 -
Tabulka 4.1: Porovnání vzdáleností	- 40 -
Tabulka 4.2: Porovnání úhlů	- 41 -

Úvod

Cílem mé práce je seznámit se s principy analýzy obrazu, jeho zpracováním, naváděním objektu pomocí zpracování obrazu a vytvořením programu, jenž bude dané zobrazení a navádění vykonávat. Součástí bude také ověření funkčnosti tohoto programu.

V první části práce se teoreticky rozebírá analýza obrazu. Jsou teoreticky popsány barevné modely, předzpracování obrazu, transformace, filtrace, segmentace obrazu a jeho převod do digitální podoby. Také je zde řešena detekce hran.

Poté se řeší návrh systému pro navádění s využitím obrazového signálu. V návrhu jsou porovnávány možnosti realizace a určuje se nejvhodnější možnost realizace naváděcího systému analyzujícím obrazový signál. K analýze je vytvořen referenční objekt, dle kterého jsou rozpoznány pohybující se objekty.

Dále je navržený systém realizován. Pro realizaci se používá jednodeskový počítač s přídavnou kamerou, která zachycuje určenou oblast. Daná oblast je analyzována. Referenční objekt je vyfocen, namapován a rozlišen v různých situacích. Upravené fotografie jsou následně pomocí neuronové sítě Tensorflow naučeny rozpoznávat jednotlivé referenční objekty. Následně lze v analyzované oblasti objekty snímat kamerou a rozpoznávat pomocí neuronové sítě. Systém zjistí, kde se v dané oblasti objekt nachází a zapíše jej do programu. Přidělí mu název dle tabulkových hodnot a navede jej směrem k předem vytyčenému cíli. Vypočítá také vzdálenost objektu a úhel natočení vůči cíli. Systém umí rozpoznat více objektů v reálném čase. Snímat analyzovanou oblast lze kamerovým modulem či USB kamerou připojenou k jednodeskovému počítači. Součástí realizace práce je tedy také porovnání obou možností snímání a na základě rychlosti a kvality rozpoznání obrazu určení, která metoda je výhodnější. Použitý programovací jazyk je Python.

V následující části práce jsou zveřejněny výsledky měření rychlosti a kvality snímání obrazu a také porovnání mezi metodami snímání. Je zde také určena verifikace rychlosti snímání, přesnosti vypočítané vzdálenosti objektu od cíle a také úhlového natočení od cíle. Nakonec je verifikována úspěšnost rozpoznání objektu.

Nakonec jsou zhodnoceny dané výsledky práce a jejich úspěšnost. Zde jsou vypsány úspěšné části práce a části, které lze vylepšit.

V dnešní době se čím dál více dostávají do povědomí společnosti pojmy automatizace a robotizace výroby. Pokrok v oblasti techniky je zřejmý a my se mu musíme čím dál rychleji přizpůsobovat. Naváděcí systém, který se řídí díky obrazovému signálu, je mým teoretickým příložením ruky k dílu této skutečnosti, navíc problematika mé bakalářské práce je nyní velmi žádaná. Při dnešním velkém rozmachu chytrých automobilů by jistě nebyl na škodu komerční systém parkování aut v domácí či průmyslové garáži, za účelem ušetření co nejvíce prostoru. Nebo využití ke stvoření hry, ovládané dvěma hráči, kdy by auta reálně existovala v jedné oblasti, ovšem hráči připojeni skrze internet, by je mohli ovládat z různých koutů světa.

1 Rozbor analýzy obrazového signálu

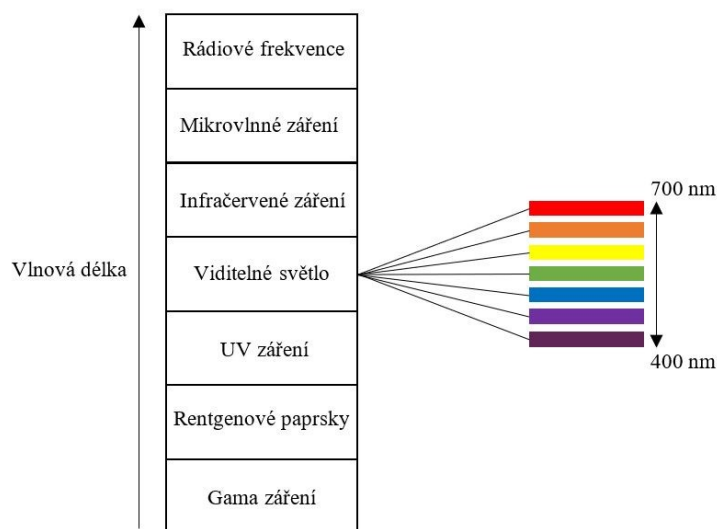
Součástí obrazového signálu, jak už název napovídá, je obraz a signál. Práce teoreticky objasní nejprve obraz a poté jeho přesun v podobě signálu. Také jsou zde vypsány možnosti zpracování obrazu, mezi které platí segmentace, transformace a filtrování obrazu.

1.1 Rozbor obrazu

Obraz je zobrazení nebo záznam vnější podoby věcí, předmětů či lidí. Z pohledu fyziky je obraz vytvářen a do lidského oka přenášen pomocí světelných paprsků. Tyto paprsky nesou informaci o vzhledu předmětu. Paprsky vznikají například na slunci a odrazem od daného předmětu vytváří informaci o jeho vzhledu. Barvu dané části určuje elektromagnetické vlnění. Elektromagnetické vlnění je charakterizováno vlnovou délkou λ [m]. Určuje se pomocí rovnice 1.1 kdy c [$\text{m} \cdot \text{s}^{-1}$] je fázová rychlost a f [Hz] je frekvence.

$$\lambda = \frac{c}{f} \quad (1.1)$$

Ne všechny elektromagnetické vlny lze vidět lidským okem. Naopak mnoho z nich vidět nelze, můžeme je však pocítit třeba na kůži. Viditelné záření se pohybuje cca od 400 do 700 nm, ovšem u každého člověka se tato čísla mohou mírně lišit. Na Obrázku 1.1 vidíme v jakém rozpětí jsou jednotlivé barvy a jak jdou za sebou a v Tabulce 1.1 můžeme vidět rozpětí frekvencí a délek u jednotlivých rozdělení vln. [7]



Obrázek 1.1: Rozdělení dle vlnové délky

Tabulka 1.1: Rozdělení spektra elektromagnetických vln [7]

Český název vln	Frekvence	Vlnová délka	Anglické označení
Extrémně dlouhé vlny	0,3 - 3 kHz	103 - 102 km	Extremely Low Frequency (ELF)
Velmi dlouhé vlny	3 - 30 kHz	102 - 10 km	Very Low Frequency (VLF)
Dlouhé vlny (DV)	30 - 300 kHz	10 - 1 km	Low Frequency (LF)
Střední vlny (SV)	0,3 - 3 MHz	1 - 0,1 km	Medium Frequency (MF)
Krátké vlny (KV)	3 - 30 MHz	100 - 10 m	High Frequency (HF)
Velmi krátké vlny (VKV)	30 - 300 MHz	10 - 1 m	Very High Frequency (VHF)
Ultra krátké vlny (UKV)	0,3 - 3 GHz	1 - 0,1 m	Ultra-High Frequency (UHF)
Mikrovlny	3 - 30 GHz	100 - 10 mm	Super High Frequency (SHF)
Mikrovlny	30 - 300 GHz	10 - 1 mm	Extremely High Frequency (EHF)
Infračervené záření	10 ¹⁰ - 10 ¹⁴ Hz	1 mm - 1 μm	Infra-Red (IR)
Viditelné záření	10 ¹⁴ Hz	400 - 700 nm	Visible (VIS)
Ultrafialové záření	10 ¹⁴ - 10 ¹⁶ Hz	400 - 10 nm	Ultra Violet (UV)
Rentgenové záření	10 ¹⁶ - 10 ¹⁹ Hz	10 - 0,1 nm	X-Rays
Gama záření	10 ¹⁹ - 10 ²⁴ Hz	10 ⁻¹⁰ - 10 ⁻¹⁴ m	Gamma Rays

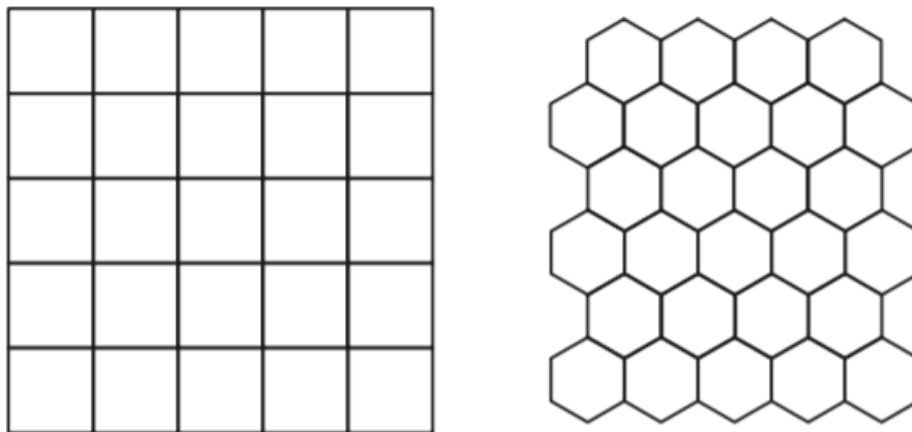
1.2 Rozbor signálu

Obraz se vytváří snímáním určité oblasti, která se přenáší signálem v podobě informací. K tomuto snímání je použito zařízení s možností snímání jasové složky. Kamera je využita kvůli možnosti práce nejen s fotografií, ale také s videem v reálném čase. Snímání se chápe jako převod optické veličiny na elektrický signál, který je spojitý v čase i úrovni. Lze jej chápat jako dvoudimenzionální signál. Vstupní veličinou tohoto signálu je jasová složka nasnímaná kamerou.

1.2.1 Digitalizace signálu

Přenos z reálného spojitého světa do počítače je prováděn pomocí diskretizace. Důvodem je, že následné zobrazení nelze provést spojitě a také protože máme omezenou paměť počítače. Obrazový signál se tedy zaznamenává pouze v konečném počtu bodů. Tyto vizuální kousky dat, jež jsme zdiskretizovali, nazýváme pixely. Podle počtu pixelů můžeme určit rozlišení námi převedeného obrazového signálu. Rozlišení je uváděno v jednotkách DPI (Dots per inch), což v doslovném překladu znamená „počet bodů na jeden palec“. Obrazové informace signálu jsou ukládány do paměti, takže je omezen nejen počet pixelů, ale také barevná hodnota nemůže být nekonečně velká. Množství barev, kterých může každý pixel nabýt, je dáno barevnou hloubkou. Barevná hloubka je obvykle zadávána v bitech a počet odstínů je 2^n . Při 8 bitové barvě to tedy bude 256 odstínů. [1]

Při digitalizaci je důležitou součástí také volba vzorkovací mřížky. Nejčastěji používané jsou mřížky čtvercové a hexagonální. Obrázek 1.2 znázorňuje právě tyto dvě mřížky. Výhodou čtvercové mřížky je, že vychází z konstrukce většiny snímacích prvků a také je velmi snadno realizovatelná. Mezi nevýhody patří především zhoršené měření vzdáleností a spojitosti objektů. Na druhou stranu hexagonální mřížka mnoho z těchto problémů řeší, ovšem není vhodná pro operace, jako je např. Fourierova transformace. [1]



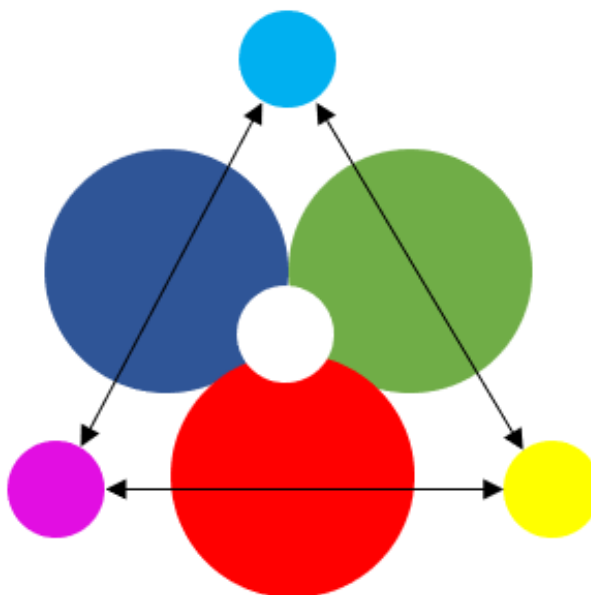
Obrázek 1.2: Čtvercová a hexagonální vzorkovací mřížka

1.2.2 Barevné modely

Barevný model využívá základní barvy a jejich mísení do barvy výsledné. Barva světla totiž může být dána i jedinou vlnovou délkou viditelného záření. Běžná je ovšem spíše směs mnoha záření různých vlnových délek. Mohou vzniknout i směsi celých spojitých spekter. Je mnoho kombinací, kterými mohou spektra vznikat. Většina spekter navíc není rozpoznatelná lidským okem. Barevné modely pak popisují v rámci daného prostoru především relativní poměry jednotlivých bodů snímku vzájemně mezi sebou, než aby měřili jednotlivé body obrazu mezi sebou. Ideální je se snažit v barevných modelech co nejvíce přiblížit opravdovým barvám. Běžně používané modely jsou ty, u kterých je zvolen vhodný kompromis mezi přesností podání barevného dojmu a složitostí konkrétního modelu a obtížností jeho použití. [8]

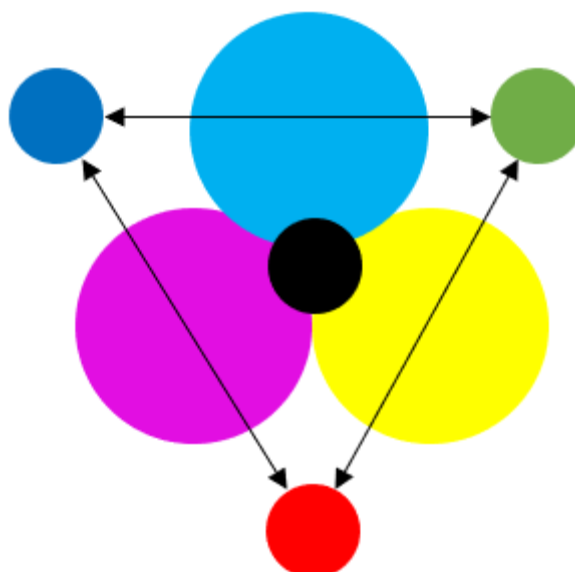
Existují dva druhy míchání barev. Jsou to aditivní míchání barev a subtraktivní míchání.

Aditivní míchání barev je způsob, kdy se jednotlivé složky barev sčítají a vytváří světlo větší intenzity. Tato intenzita se poté rovná součtu intenzit jednotlivých složek. Základní barvy jsou červená, modrá a zelená. Tento způsob je používán u monitorů a televizí.



Obrázek 1.3: Aditivní míchání barev

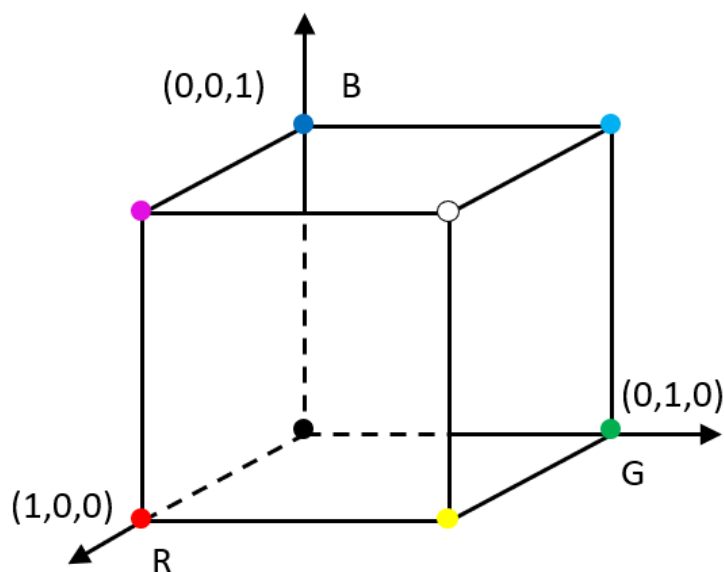
Subtraktivní míchání barev je způsob, kdy s každou další přidanou barvou se ubírá část původní barvy. Jednotlivé světelné vlny procházejí barevnými vrstvami a jsou po přidání pohlcovány. Výslednou barvu tvoří ty složky, které nebyly odstraněny. Základní barvy jsou azurová, purpurová a žlutá. Tento způsob je používán u tiskáren.



Obrázek 1.4: Subtraktivní míchání barev

Barevný model RGB

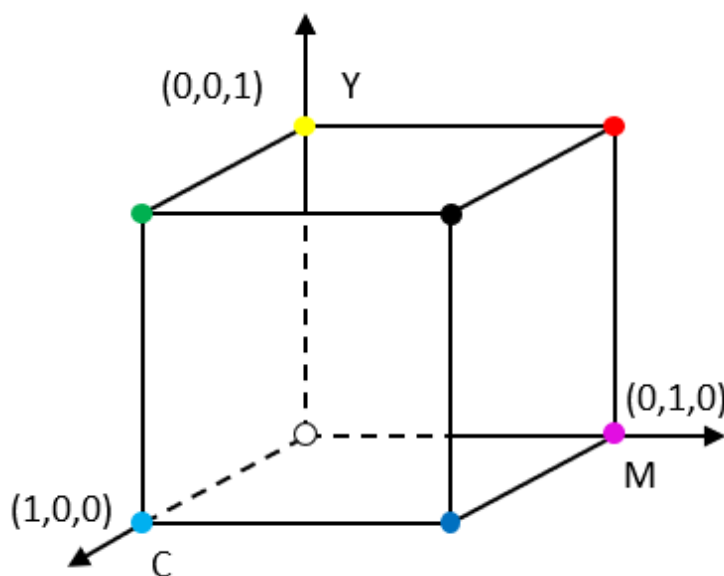
Řadí se mezi aditivní modely. Zkratka RGB znamená red, green, blue, takže vychází ze tří barev červené, zelené a modré. Lidské oko je totiž závislé na těchto třech barvách a jejich kombinací získáme poté všechny ostatní nám známé barvy. Dá se zobrazit jako 3D krychle, kdy poměr jednotlivých barev bude udávat barvu výslednou a vzdálenost od počátku zase jas. Černá barva se zde nachází v bodě $[0;0;0]$. Při maximálním jasu všech složek dosáhneme barvy bílé. [8]



Obrázek 1.5: 3D RGB model

Barevný model CMY/CMYK

Řadí se mezi subtraktivní modely. Zkratka znamená cyan, magenta, yellow a key. Při překladi do češtiny zjistíme, že se znovu jedná o názvy barev, tedy azurová, purpurová, žlutá a černá. S rostoucí hodnotou jednotlivých složek dochází k zúžení barevného spektra. Je tedy postupováno nanášením pigmentů na bílý podklad. Je logické, že hlavní využití tohoto modelu je u tiskáren. Ačkoliv teoreticky by se černá barva měla vytvořit subtraktivním smísením 100 % všech příměsí, ve skutečnosti se tomu tak neděje. Proto se k základnímu modelu CMY přiřazuje ještě barva černá a vzniká CMYK. Obvykle se hodnoty jednotlivých složek udávají v procentech. Zde při nulovém jasu všech barev získáme barvu bílou a při maximálním naopak barvu černou. [8]



Obrázek 1.6: 3D CMY model

Barevný model HSV

Nejsou zde základní barvy, jako u předchozích modelů, ale jedná se o kombinaci hue, saturation a value (odstín, saturace a hodnota). Lze také najít zkratky jako HSI či HSB, kdy je pouze value zaměněna za intensivity či balance. Hlavní myšlenku to ve své podstatě nemění, kdy se jedná o kombinaci tónu, sytosti a jasů. Tón je udáván ve stupních, od 0° do 360° (0° resp. 360° - červená, 120° - zelená, 240° - modrá). Sytost určuje přidanou bílou složku a jas určuje, kolik světla barva odrazí. [8]

1.3 Předzpracování obrazu

Pokud dojde k úspěšnému získání obrazového signálu a jeho digitalizaci, máme k dispozici digitální obraz pozorované scény. Obraz ovšem může být zkreslen či jinak porušen, díky způsobu snímání nebo kvůli nevhodným podmínkám při snímání. Pokud je znám charakter zkreslení, je možné tuto chybu opravit pomocí korekcí.

Chyby v obrazu po digitalizaci mohou být opraveny Jasovou či Geometrickou transformací, či filtrací. Jasové transformace máme globální, lokální a bodové, zatímco filtrace jsou pouze lineární či nelineární. Na různé chyby v obrazu se hodí využít jiný způsob úpravy těmito prostředky.

1.3.1 Jasové transformace

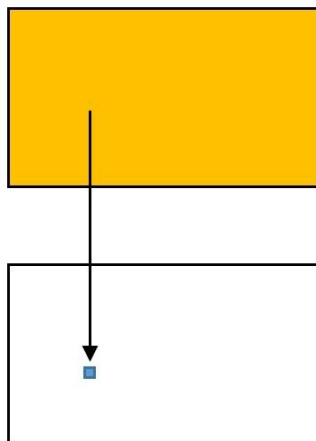
Při snímání by zařízení mělo mít v ideálním případě stejnou citlivost na všechny body v obraze, avšak tento požadavek není vždy splněn. V optických soustavách, bývá světlo procházející dále od optické osy, zeslabováno. Špatné snímání může nastat také při prachu na skle snímače či při špatném osvětlení snímaného prostředí. Pokud tyto poruchy jsou systematické, což znamená že nejsou náhodné, dají se jednoduše eliminovat. Zvolíme-li totiž vhodnou funkci, bude tuto systematickou chybu vyrušovat během celého procesu snímání. [5]

Transformace působí na vstupní obraz a vytvářejí výstupní obraz stejných parametrů tzn. rozlišení či bitová hloubka se nezmění. Ovšem při transformaci nedochází k rozpoznání objektů. K tomu dochází až při segmentaci, kterou jsem uvedl v bodě 1.4. Máme tři druhy jasové transformace a to globální, lokální a bodovou.

Globální transformace

Při globální jasové transformaci má transformovaný nový pixel jasovou hodnotu vypočítanou z hodnot celého obrazu. Příkladem globální transformace je integrální obraz či Fourierova transformace. Znázorníme ji rovnicí 1.2

$$\Omega(I) \xrightarrow{T} f(x, y) \quad (1.2)$$

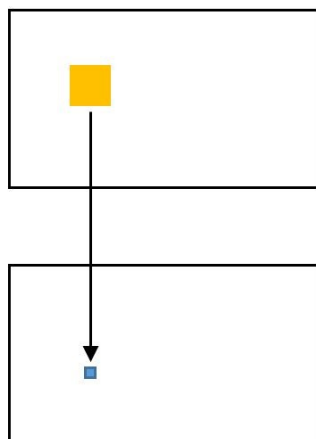


Obrázek 1.7: Globální transformace

Lokální transformace

Při lokální jasové transformaci má transformovaný nový pixel jasovou hodnotu vypočítanou z hodnot lokálního okolí pixelu.

$$\Omega(x, y) \xrightarrow{T} f(x, y) \quad (1.3)$$



Obrázek 1.8: Lokální transformace

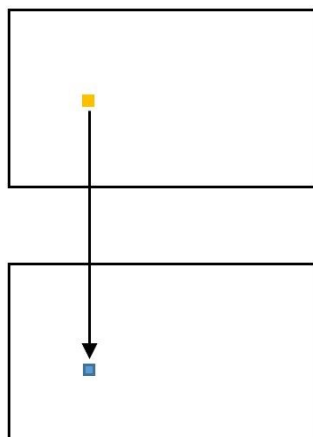
Hodnoty na lokální transformaci bereme hodnoty z lokálního okolí pixelu, se kterými pak musíme pracovat. Používané metody jsou třeba prostý průměr, modus či medián. Pokud bychom lokální transformaci chtěli použít s pomocí prostého průměru, musíme výpočetně řešit konvoluci, alias součinem hodnot. Příkladem je lokální filtrace či zvýraznění rysů např. hran.

$$f(x, y) = g(x, y) * h(x, y) \quad (1.4)$$

Bodová transformace

Transformovaný nový pixel při bodové jasové má jasovou hodnotu vypočítanou jen z hodnoty daného pixelu. Zde je příkladem například jasová korekce, převodní charakteristika či ekvalizace histogramu.

$$f'(x, y) \xrightarrow{T} f(x, y) \quad (1.5)$$



Obrázek 1.9: Bodová transformace

Histogram je sloupkový graf, udávající informaci o výskytu jednotlivých jasových úrovní v obraze. Na vodorovné ose zachycuje intervaly, do kterých jsou jistá data rozčleněna a na svislé ose počet výskytů těchto dat v daném intervalu. U obrazu to znamená, že na ose x se vyskytují úrovně jasové stupnice a na ose y je pak zobrazen jejich počet. Histogram je obvykle jedinou globální informací o obraze. Jelikož navíc udává pouze informaci o počtu jednotlivých jasových hodnot, tak pomocí něj není možné zjistit např. pohyb objektu na snímané scéně, protože histogram se nezmění. Při ekvalizaci histogramu transformujeme jasovou stupnici obrazu tak, aby ve výstupním obraze byly jasové složky zastoupeny rovnoměrně. Zvyšuje se tím např. kontrast pro pozorovatele, či nastává zlepšení podmínek pro další zpracování obrazu. [5]

1.3.2 Geometrická transformace

Vstupní obraz je občas nutné taky upravit pomocí geometrických metod. Mezi tyto metody patří otočení, změna velikost, posunutí, rozšíření atd. Také umožňují odstranit geometrické zkreslení vzniklé při pořízení obrazu, např. korekce geometrických vad objektivu kamery. Většina úloh zpracování obrazu pomocí geometrické transformace lze provést v rovině, alias 2D.

1.3.3 Filtrace

Při úpravách kvality obrazu jsou důležité různé druhy filtrace. Tyto filtrace mohou upravit zrnitost záznamu a také zvýraznit jiné objekty v obraze. Vlastně se jedná o úpravu intenzity obrazového bodu s vazbou na jeho okolí, které přitom může být naprosto rozličné. Jsou možnosti obraz upravovat pomocí lineární či nelineární filtrace. [9]

Lineární filtrace

Intenzita upravovaného bodu u lineárních filtrů je rovna součtu součinů intenzit bodů v okolí a příslušných váhových koeficientů z matice váhových koeficientů. Součin nazýváme konvoluce. Lineární filtry, např. dolní propust, slouží především k odstranění vysokých prostorových frekvencí intenzit v obraze. Povede se nám touto metodou potlačit nežádoucí šum, ale také potlačit detaily v obraze. Tato filtrace se používá především k odstranění zrnitosti v obrazech. [5]

Mezi metody lineární filtrace patří vyhlazování průměrováním či Gaussovo vyhlazování.

Nelineární filtrace

Nelineární metody dokáží částečně eliminovat hlavní nevýhodu lineárních filtrů, a to rozmazávání hran. Jejich princip je, že se snaží v okolí každého bodu nalézt pouze tu část pixelů, která má podobnou hodnotu jasu jako zpracováváný bod. Jen podobné pixely budou později využity pro zpracování daného bodu. [5]

Patří zde např. metoda rolující masky.

1.4 Segmentace

Jedná se o rozpoznání objektů na určitém obraze. Přichází na řadu po předzpracování obrazu a cílem je extrakce rozpoznávaných objektů. Objekty jsou separovány od pozadí. Jedná se o nejsložitější část zpracování obrazu, vzhledem k tomu, že objekty se mohou překrývat částí či se dotýkat hranami a nelze je pak tak jednoduše rozpoznat. Proto na pomoc segmentaci používáme thresholding alias prahování.

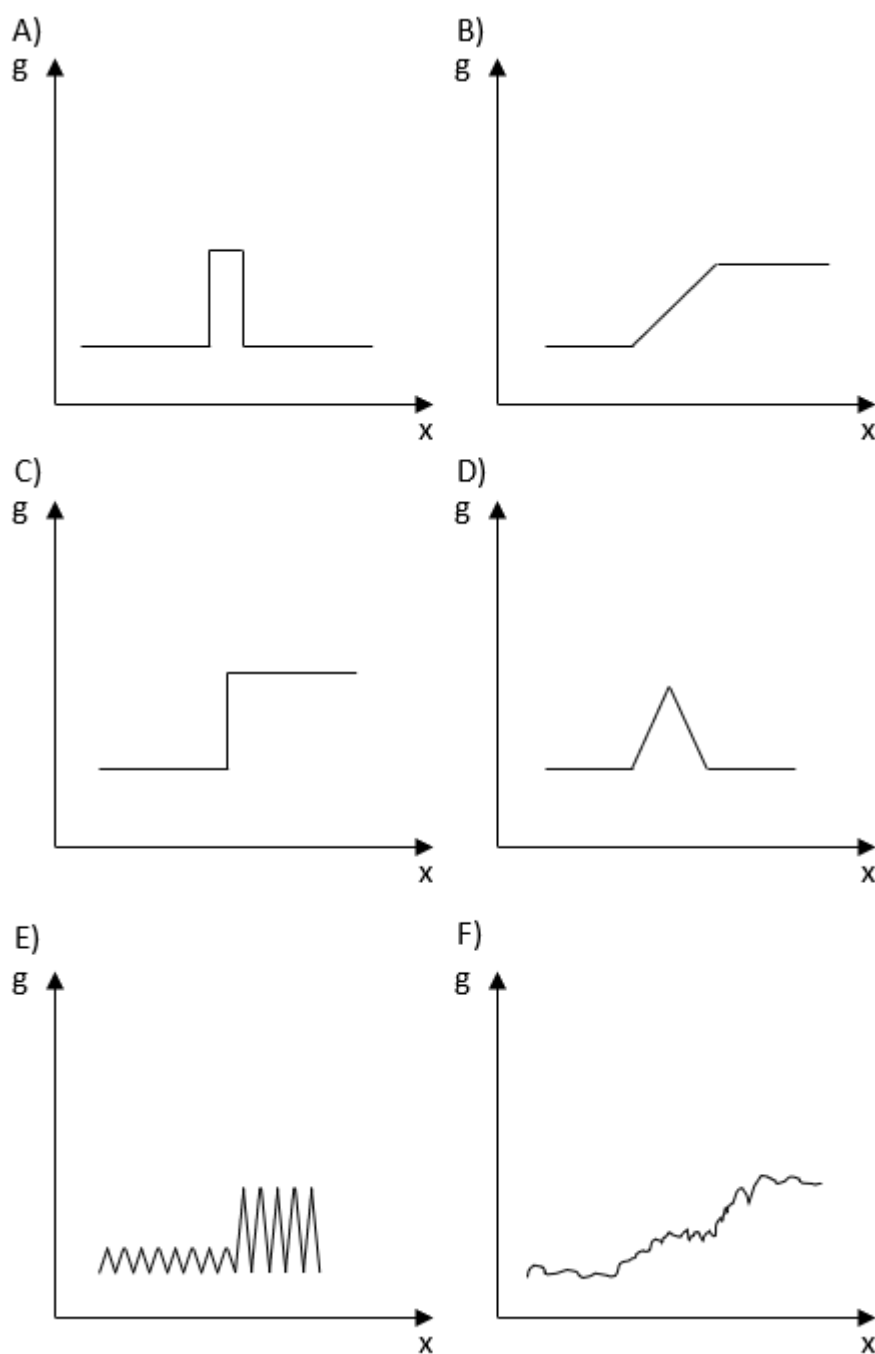
Thresholding se používá pro úpravu obrazu s odstíny šedé barvy na černé a bílé pixely. Musí se nalézt optimální úroveň thresholdingu, aby po této operaci zůstaly viditelné pouze potřebné hrany a ostatní zmizely. V dalším kroku jsou pak převedeny hodnoty větší, než je daná úroveň, na bílou. Zbytek se převede na černou barvu. Úroveň thresholdingu zjistíme několika způsoby. Tím nejjednodušším je vybrat práh pomocí histogramu. Dalším způsobem je nalezení úrovně automaticky podle automatického nalezení úrovně či Otsuovy metody. [2]

1.5 Detekce hran

Nejdůležitější informace u algoritmů, které řeší detekci hran v obraze, jsou o hranách v obraze. Tyto algoritmy se sestávají ze dvou částí, v jedné se hrany v obraze nalézají a v druhé se tyto nalezené hrany segmentují a hledá se hranice vzniklých segmentů. Metod na detekci hran máme více a dělí se dle rozdílné interpretace a zacházení s výslednou segmentací. Každá metoda vytváří následně díky segmentaci mapu hran, což lze považovat jako nový obraz. [6]

1.5.1 Hrana

Jedná se o oblast v obraze, ve které při analyzování dochází k velké změně intenzity. Druhy hran lze popsat jednoduchými grafy. Grafy ukazují funkci, jež ukazuje hodnoty intenzity pro daný pixel v daném bodě.



Obrázek 1.10: Grafy různé intenzity detekované hrany

Většina druhů hran lze řešit pomocí první či druhé derivace. Ovšem je zde problém při typu hrany, která je na hranici s jinou hranou či jinou texturou. Protože derivace dokáží dobře nalézt hrany uvnitř textury, nikoliv hranici mezi dvěma texturami. Na takové hrany se proto používá srovnání se vzorem či statistická metoda. Srovnání se vzorem (template matching) je proto také použito pro tento systém.

1.5.2 První derivace

Detekce hran první derivací funguje na principu, že v místech, kde se hrana nachází, dochází k největší změně intenzity. V ostatních oblastech tedy není žádná změna, a tím pádem i první derivace je rovna nule. Jedná se relativně rychlý výpočet, který není složitý aplikovat. Jeho problémem je ovšem vytváření tlustých hran, což zapříčiňuje následnou nepřesnost nalezené hrany. [1]

1.5.3 Druhá derivace

Není-li nutné znát směr a velikost hledaných hran, je dobré použít druhou derivaci. Jedná se o způsob nalezení hran tak, že je z první derivace zjištěno, kdy nastává změna intenzity tím, že má nějakou hodnotu, a tak druhá derivace má hodnotu nulovou. Nalezené hrany jsou tedy v místech, kde druhá derivace prochází nulou. Jsou tak nalezeny pouze pozice hran, nikoliv jejich směr a velikost. [1]

1.5.4 Srovnání se vzorem

Základním prvkem je jistá maska, která se porovnává s právě kontrolovaným místem. Najde-li algoritmus dostatečnou shodu s maskou, dojde k nalezení hrany. Tato metoda je velmi účinná vůči obrazovému šumu, který závisí na velikosti masky. Také lze rozeznat hrany na hranici dvou textur, což je největší problém již zmíněné první a druhé derivace. Slabinou je závislost na natočení obrazu a také časová náročnost probíhající algoritmu, který musí porovnávat masku s kontrolovaným místem. [13]

2 Návrh systému

Cílem návrhu je vytvořit naváděcí systém analyzující obrazový signál, který umožňuje snímání obrazu v reálném čase analyzovaném na jednodeskovém počítači. Výsledkem dané analýzy je rozpoznání referenčního objektu či objekty, vypočítaná vzdálenost daných objektů od jejich cíle, úhel natočení nalezených objektů a také směr, kterým se mají vydat. Vše v reálném čase za použití jednodeskového počítače na zpracování. Analýza obrazu je provedena přes OpenCV a Tensorflow knihovnu za použití programovacího jazyku Python. OpenCV je knihovna, určená k manipulaci s obrazem a zaměřuje se na počítačové vidění a zpracování obrazu v reálném čase. Tensorflow je přednastavená matematická knihovna, jejíž součástí je neuronová síť s možností strojového učení.

2.1 Softwarové řešení

Základním bodem návrhu naváděcího systému je instalace knihoven a shromáždění podkladů k analýze obrazu. Je nutné začít s nainstalováním či aktualizací softwaru na námi použitým hardwarovém prostředí. U této práce se jedná o jednodeskový počítač Raspberry Pi 3B+ a stolní počítač. K výslednému produktu naváděcího systému pomocí analýzy obrazu, který je prováděn čistě na jednodeskovém počítači je nutné dodat natrénovaný model. Tento model nelze natrénovat na počítači s malou výpočetní silou, takže je nutné si neuronové síť připravit jak na jednodeskovém počítači na následnou aplikaci, tak i na stolní počítač kvůli přípravě. Je nutné myslet na rozdíly jednotlivých přístrojů, hlavně ve výkonu a operačním systému, kdy na stolním počítači se pracuje s Windows 10, zatímco na Raspberry Pi je použit Raspbian Stretch, Linuxový operační systém.

O komunikaci mezi stolním počítačem a jednodeskovým počítačem se stará komunikační protokol SSH či grafické rozhraní VNC. Já jsem si vybral SSH za pomoci aplikace MobaXterm. Komunikaci je nutno povolit na jednodeskovém počítači a následně se přes přidělenou IP adresu připojit.

2.2 Rozbor problematiky navádění

Naváděcí systém analyzující obrazový signál pracuje na základě zjištěných informací, které lze následně předat do naváděného objektu. K navádění objektu do cíle je nutné rozlišit, kde se objekt nachází. To se děje pomocí analýzy obrazu skrze neuronové síť a strojové učení. Objekt je nalezen pomocí předem určeného referenčního objektu, jenž je na naváděný objekt možno přidat. Při nalezení daného referenčního objektu je nutné uvést, zdali je nalezen i směrový bod objektu. Tento bod se nachází na referenčním objektu ve směru natočení objektu a díky jeho nalezení je možno směr objektu určit. Při nalezení směrového bodu objektu je také nutné rozpoznat, zdali náleží v objektu, či se jedná o chybu analýzy obrazu. Pokud se povede úspěšně nalézt jak referenční objekt, tak jeho směrový bod v oblasti objektu, je známa poloha objektu i jeho směr natočení a tyto informace je možné předat naváděnému objektu.

Na základě těchto informací je poté proveden výpočet vzdálenosti objektu od námi předem stanoveného cíle. Tato hodnota je udávána jak v celkové vzdálenosti, tak ve vzdálenosti na ose x a ose y. Použitá hodnota je px nebo při převedení do hodnot objektu mm. Také je vypočítán úhel, pod kterým je objekt natočen vůči cíli. Tento úhel je získán porovnáním úsečky mezi objektem a jeho směrovým bodem a objektem a cílem.

Předání zjištěných informací analýzou obrazového signálu a následným výpočtem do objektu slouží ke správnému navádění. Po předání informací dojde ke zpracování naváděným objektem a následnou změnou polohy objektu. Po změně polohy objektu dojde k novému zpracování dat naváděcím systémem analyzujícím obrazový signál a znovu zjištěné informace se opět odesílají naváděnému objektu.

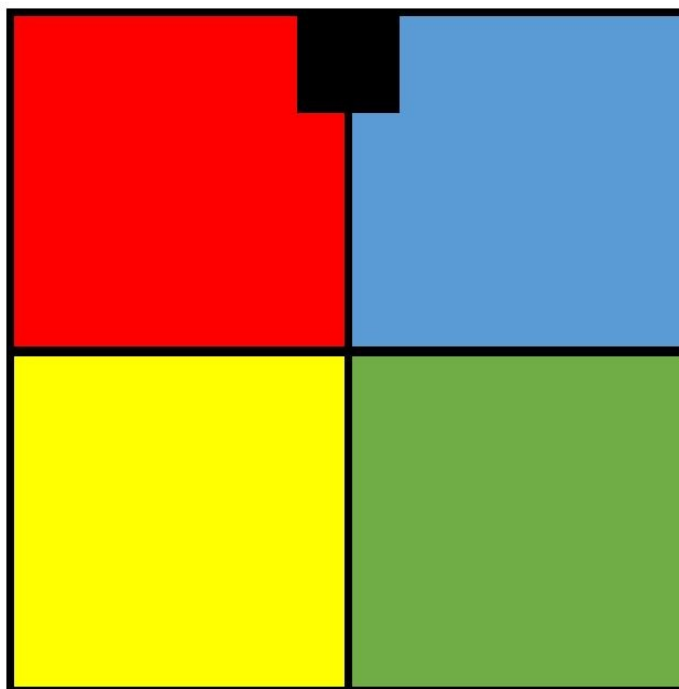


Obrázek 2.1: Vývojový diagram procesu navádění

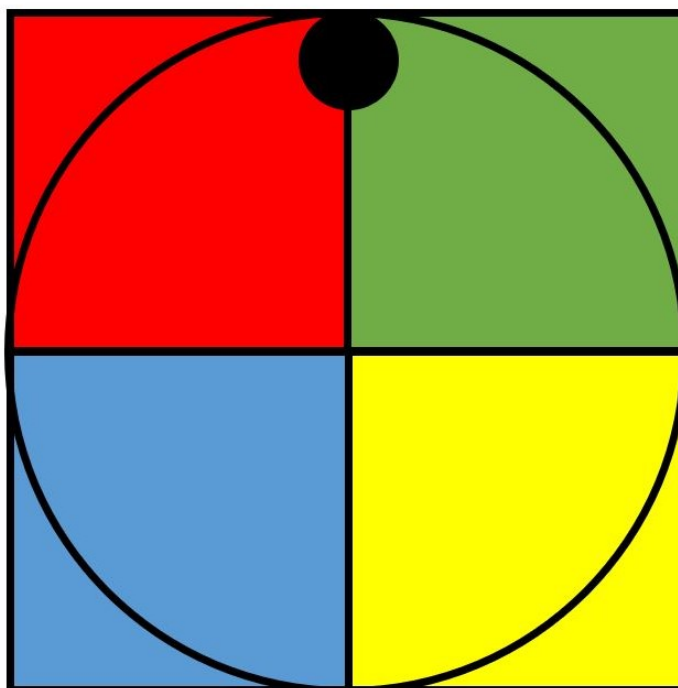
2.3 Referenční objekty

Podstatná součást navrhovaného naváděcího systému analyzujícího obrazový signál jsou referenční objekty, jež slouží k rozpoznání, ve které části se objekt v obraze nachází. Pro svou bakalářskou práci jsem zvolil dva rozlišné objekty. Liší se jak vzorem, tak pořadím barev. Důvodem je, že neuronová síť nejlépe rozeznává objekty, které jsou si co nejvíce rozdílné, ovšem rozdíl zase nesmí být tak velký, aby byla metoda dobře ozkoušena. Referenční objekt čtverec má rozměr 10x10 cm a pořadí barev zleva nahoře, vpravo nahoře, vpravo dole, vlevo dole je červená, modrá, zelená a žlutá. Zatímco referenční objekt kruh má průměr 10 cm a barvy ve stejném pořadí jsou červená, zelená, žlutá a modrá.

Oba objekty jsou vybaveny nejen čarami na rozdělení barev, ale také směrovým bodem objektu. Směrový bod objekt je černý čtverec či černý kruh na referenčním objektu, jenž slouží k rozpoznání směru, kterým daný objekt míří.



Obrázek 2.2: Referenční objekt čtverec

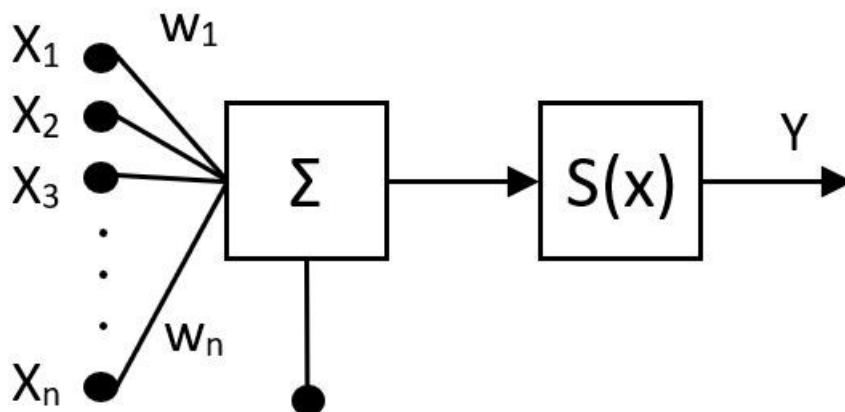


Obrázek 2.3: Referenční objekt kruh

2.4 Umělá neuronová síť

Jedná se o výpočetní model používaný při práci s umělou inteligencí a strojovým učení. Umělá neuronová síť není jednotlivý algoritmus, ale prostor, kde mnoho algoritmů strojového učení může spolupracovat a zpracovávat složitá vstupní data. Učící proces těchto sítí probíhá na základě porovnání předpřipravených příkladů a poté nepřipravených vstupních dat, která jsou vložena. Například rozpoznání obrazu funguje na základě předpřipravených obrazů. V nich je zvolený hledaný objekt rozpoznán a neuronová síť si na základě těchto informací vytvoří vlastní rozpoznávací charakteristiky, které můžeme později použít k rozpoznání daných objektů v nových obrazech. [10]

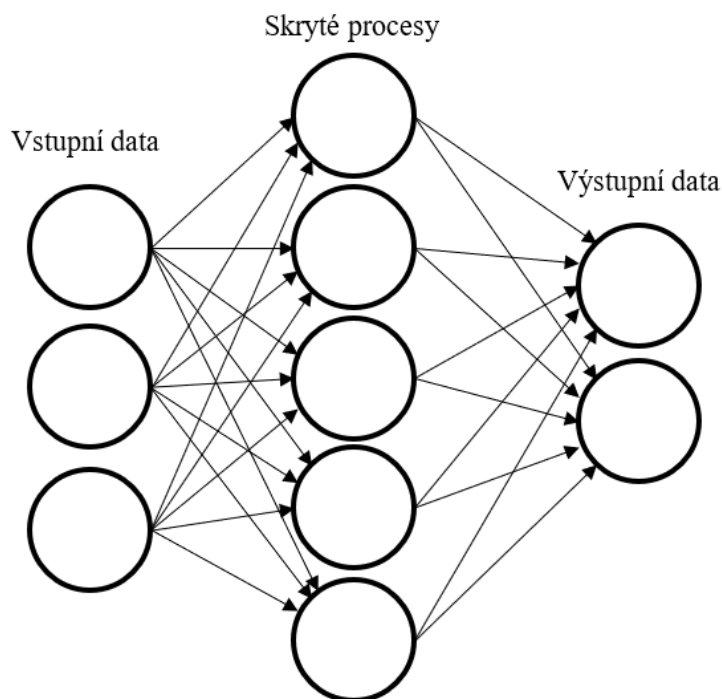
Umělá neuronová síť má základ v biologických neuronových strukturách. Struktury použité v umělých sítích se nazývají umělé neurony. Umělé neurony získávají jeden nebo více vstupních informací a vytvářejí z nich výstup. Každý ze vstupů je vážen. Vážení je proces, který zdůrazňuje konkrétní aspekty dané problematiky vůči ostatním aspektům vzhledem k výslednému produktu. Výsledky vážení se poté vloží do přenosové funkce, která teoreticky vymodeluje graf pro každý jeden vstup. [10]



Obrázek 2.4: Umělý neuronový model

$$y_k = S(x) \left(\sum_{j=0}^n w_j x_j \right) \quad (2.1)$$

V umělých neuronových sítích jsou jednotlivé umělé neurony spojeny navzájem. Stejně jako synapse v mozku, dokáží přesunout informaci z jednoho umělého neuronu do druhého. Informaci neuron zpracuje a předá ji zase dalším, na sebe napojeným umělým neuronům. Lze to vidět na Obrázku 2.5.



Obrázek 2.5: Propojení umělých neuronů v umělé neuronové síti

Běžné je, že informace přenesená mezi dvěma umělými neurony v umělé neuronové síti je reálné číslo a výstup každého neuronu je vypočítán nějakou nelineární funkcí součtu jeho vstupů. Spojení mezi neurony se nazývá hrana. Umělé neurony a hrany mají hodnotu vážení, která je v průběhu učení upravována. Tato hodnota snižuje či zvyšuje hodnotu předávané informace v bodu spojení dvou neuronů. Každý neuron má také práh, kdy při překročení jeho hodnoty není informace z předchozího neuronu předána. Umělé neurony jsou umístěné v jednotlivých vrstvách v neuronové síti. Podle jejich rozřazení určité vrstvy vykonávají různé procesy. Informace tedy proudí nejen od neuronu do neuronu, ale také musí projít přes všechny vrstvy dané umělé neuronové sítě. [10]

Neuronové síť lze využít k mnoha účelům. Nejedná se pouze o již zmíněné obrazové zpracování, ale také k práci s řečí, kdy lze rozeznávat jednotlivé jazyky, vyhledávat příspěvky na sociálních sítích dle zadaného klíče či překládání jazyků. Lze naučit neuronovou síť ovládat počítačové hry či jiné programy, ale také určovat medicínské diagnózy.

Mnou zvolená umělá neuronová síť je symbolická matematická knihovna Tensorflow. Tensorflow je open-source projekt vydávaný společností Google.

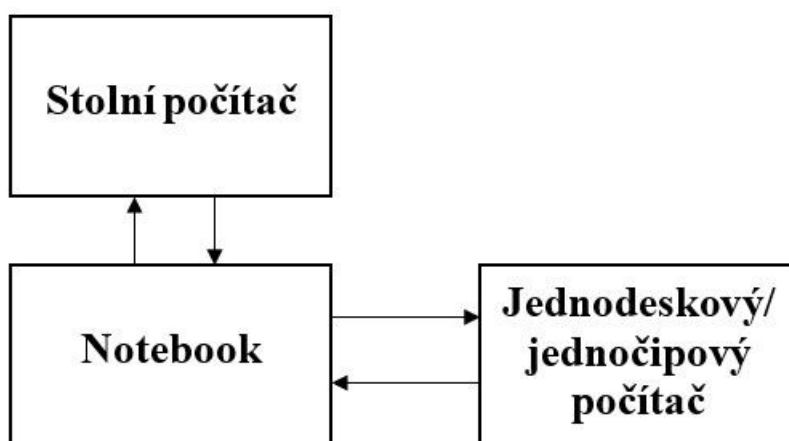
2.5 Strojové učení

Strojové učení je systém skládající se z algoritmů a statistických modelů. Tento systém je používán počítačem k specifické práci s daty, při které nemá určené přesné instrukce postupů provedení. Spíše záleží na vzorech a odvozeních původních hodnot. Učení lze proto také popsat jako změna vnitřního stavu systému, která zefektivní schopnost přizpůsobení se změnám okolního prostředí. Jedná se o podoblast umělé inteligence. [11]

Základní rozdělení algoritmů učení máme dle způsobu učení a také dle způsobu zpracování dat. Dle učení umíme rozeznat učení s učitelem, učení bez učitele, jejich kombinaci a také zpětnovazební učení. Dle způsobu zpracování dat rozdělujeme učení na dávkové a inkrementální.

2.6 Návrh hardwarových komponent naváděcího systému

Hardwarová část naváděcího systému se skládá ze dvou částí. V jedné části je obsluha a příprava systému a ve druhé provedení a aplikace systému. Na aplikaci systému je navržen jednodeskový počítač Raspberry Pi nebo jednočipový počítač Arduino. Tato část je samostatná a vykonává analýzu obrazového signálu a umožňuje působení naváděcího systému. K obsluze a přípravě systému slouží stolní počítač či notebook. Notebook je použit jako obslužná část systému, kdy při připojení přes SSH protokol lze aplikaci obsluhovat či měnit. Stolní počítač je použit z důvodu vyššího výkonu. Na vytvoření modelu rozpoznávajícího referenční objekt je nutné mít kvalitní a výkonnou grafickou kartu a procesor.



Obrázek 2.6: Návrh propojení hardwarových komponent

3 Realizace systému

Naváděcí systém na analýzu obrazového systému je realizován na jednodeskovém počítači s připojenou kamerou, kterou představuje buď kamerový modul nebo USB kamera. Z důvodu velké softwarové náročnosti je doporučeno využít také přídavný procesorový chladič na jednodeskový počítač z důvodu přehřátí. Systém je realizován pomocí učení neuronové sítě a strojového učení, na jehož základě dokáže rozpoznávat předem vybrané referenční objekty. Díky tomu zvládne zjistit vlastnosti objektu a pomocí nich úspěšně provádět navádění.

3.1 Řešení hardwarových komponent

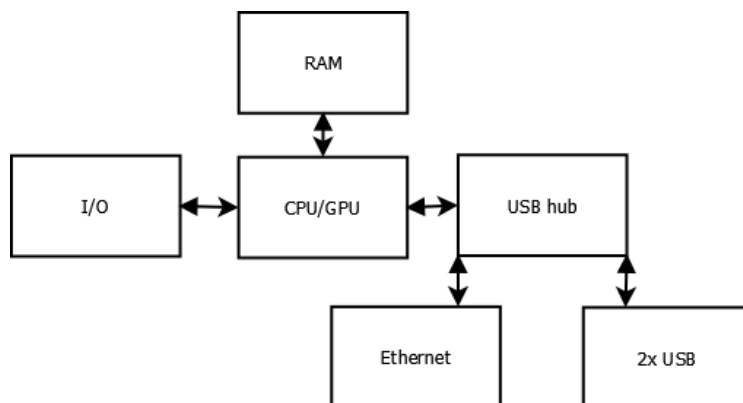
Naváděcí systém je soubor mnoha funkčních částí a je nutné vybrat správné komponenty na řešení dané problematiky. Konečný vybraný systém se skládá z několika hardwarových komponent, konkrétně z připojeného notebooku, jednodeskového počítače Raspberry Pi, kamerového modulu, referenčního objektu a kvádrového stojanu, jež drží kamerový modul nad plochou, která je sledována. Jsou zde porovnány jednodeskové počítače Raspberry Pi a Arduino, jelikož se jedná o dvě základní teoretická možnosti řešení a také kamerový modul Raspberry Pi a USB kamera.

3.1.1 Jednočipový počítač Arduino

Arduino je jedním z průkopníků mezi malými jednočipovými počítači. Vývoj se nastartoval v roce 2005, kdy se vývojáři z italského Interaction Design Institute ve městě Ivrea snažili vytvořit levný vývojový set pro studenty. Arduino se mezi studenty uchytilo a vývojáři jej rozšířili do všech oblastí, nejen k studijním účelům. Jedná se o Open Source projekt, takže většina návodů a schémat jsou veřejně dostupná k bezplatnému použití. Arduino je založeno na programovacím jazyku Processing. Na rozdíl od Raspberry Pi není zamýšleno jako plnohodnotný stolní počítač. Nelze řídicí program vytvářet přímo v Arduinu, musíme jej vytvořit skrze stolní počítač či notebook a do Arduina teprve nahrát. Díky tomu má malou spotřebu a hodí se k ovládání dronů, robotů či jiných autonomních zařízení.

3.1.2 Jednodeskový počítač Raspberry Pi

Jedná se o malý jednodeskový počítač. Vyvinula jej britská nadace Raspberry Pi Foundation v roce 2012 za cílem podpory výuky informatiky ve školách a seznámit studenty s tím, jak mohou různá zařízení např. mikrovlnná trouba či automatická pračka, být řízeny. Primární operační systém je Raspbian. Obsahuje konektor pro vývod obrazu HDMI, přes USB je možné připojit klávesnici a myš. Na rozdíl od počítače Arduino lze Raspberry Pi použít nejen k ovládání různých zařízení (pomocí GPIO kontaktů), ale i k vývoji určitých aplikací. Také jej můžeme použít jako multimediální přehrávač videa nebo hudby nebo i jen pro přístup k internetu (s potřebnými distribucemi).



Obrázek 3.1: Schéma Raspberry Pi

Z důvodu vyššího výkonu, možnosti ovládání přímo z konzole přes grafické rozhraní a také přijatelné cenové relaci pro pozdější možnost využití, je pro naváděcí systém na analýzu obrazu vybrán jednodeskový počítač Raspberry Pi, konkrétně jeho variace 3B+. Jedná se o nejnovější verzi, v době vypracování této bakalářské práce, z roku 2018. Výraznou změnou je procesor 1.4 GHz Cortex-A53 a síťové propojení skrz Bluetooth a Wi-Fi. Uložištěm operačního systému a následných programů a knihoven je MikroSDHC karta, pro tuto práci byla pořízena karta s 32 GB pamětí. Ovládání je možné skrze vlastní grafické rozhraní Raspbian nebo přes komunikační protokol SSH či grafický program VNC z PC.

3.1.3 USB kamera

USB kamera je použita kvůli své univerzálnosti. Lze ji připojit ke stolnímu či přenosnému počítači a taktéž k jednodeskovému počítači jako je Raspberry Pi. Použitá kamera v případě mé bakalářské práce je A4Tech PK-910H. Jedná se o model od společnosti A4Tech, s FullHD rozlišením, používaný hlavně pro online komunikaci. Rozlišení USB kamery je 1920x1080 px a zvládne snímat 30 fps. Součástí softwaru kamery je také automatické zaostřování, které ovšem pro svou bakalářskou práci nevyužívám. Připojení k jednodeskovému počítači zajišťuje rozhraní USB 2.0.

3.1.4 Kamerový modul Raspberry Pi

Kamerový modul je součástí snímající danou oblast za účelem rozpoznání referenčního objektu. Oblast je následně analyzována v jednodeskovém počítači. Modul má rozlišení 5 Mpx a podporuje video v kvalitě 1080p30, 720p60 a 640x480p60/90. Připojení je zajištěno flex kabelem, dlouhým 15 cm, který se připojuje k CSI portu na Raspberry Pi. Na základě dostupných komponentů byl vybrán Raspberry Pi kamerový modul V1.

3.1.5 Realizace testovacího přípravku

Celé řešení hardwarových komponent je umístěno na kvádrovém stojanu, jež má za úkol držet jednodeskový počítač s připojeným kamerovým modulem a umožňovat bezproblémové snímání plochy na které se referenční objekt hledá. Plocha pod kvádrovým stojanem je bílá, aby nedocházelo k problémům snímání, například se špatným rozpoznáním referenčního objektu. Rozpoznání je jednodušší díky detekci černých hran na referenčním objektu oproti bílé ploše.



Obrázek 3.2: Kvádrový stojan na tetovací přípravek

3.2 Implementace učicího procesu neuronové sítě

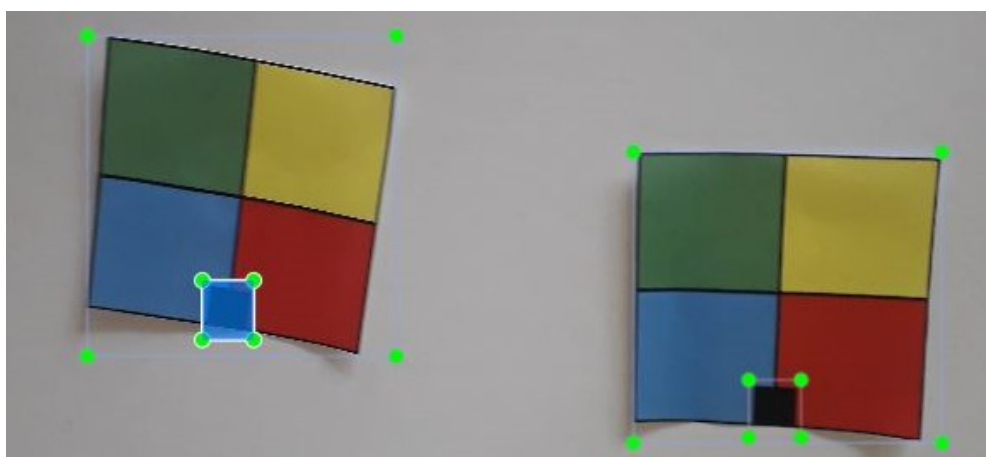
Učící proces neuronové sítě je základem k rozpoznání vlastního referenčního objektu při analýze obrazového signálu. Je tedy nutné, zvolit si referenční objekt či objekty, na které bude neuronová síť naučena reagovat a rozpoznávat. Učící proces má za úkol vytvořit model, dle kterého lze referenční objekt úspěšně rozeznat v určitých podmínkách.

Základ učicího procesu je získání vstupních dat z referenčního objektu. Získání těchto dat lze pomocí fotografií referenčních objektů v prostředí, ve kterém jej následně naučený model bude rozpoznávat při své práci. U mé bakalářské práce je nutné naučit neuronovou síť rozpoznávat referenční modely v kvádrovém čtverci, který má jako dno bílou plochu, takže vytvořené fotografie jsou na této ploše. Je nutné mít více vstupních fotografií, nejlépe z více úhlů, ze všech stran natočení, z různých denních dob a následně změně světelných paprsků. Naváděcí systém například špatně rozeznává objekty při svícení žárovek s jinou teplotou chromatičnosti, nežli má slunce či při slabém světelném nasvícení. Tento problém lze vyřešit poskytnutím vstupních fotografií při takovém záření či vícečetným trénováním modelu. Fotografie se následně musí upravit, aby neuronová síť věděla, kde přesně náleží hledaný objekt a mohla tak vytvořit model pro námi zvolené referenční objekty. Úpravu

fotografií zajišťovala aplikace LabelImg. Pomocí této aplikace se jednotlivým objektům na fotografiích přiřadili třídy, pomocí nichž následně rozdělujeme hledané objekty. Tato aplikace umožňuje zvýraznit polohu hledaného objektu v obraze a vytvořit následně stejně pojmenovaný soubor typu xml, do kterého zaneseme hodnoty x a y pozice jednotlivých objektů a jejich námi určených tříd. Celkový počet použitých fotografií byl 153 a stejný počet xml souborů. Tyto fotografie a soubory byly následně rozděleny do dvou složek, nazvaných test a train, v poměru 20 ku 80 procentům.



Obrázek 3.3: Fotografie objektů v prostředí

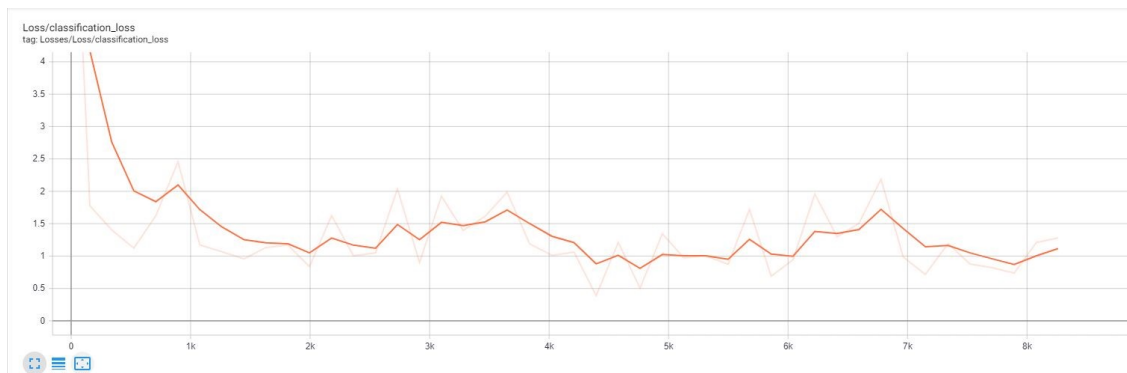


Obrázek 3.4: Fotografie objektů v prostředí se zvýrazněním objektů

Následně, po vložení upravených fotografií je nutné připravit neuronovou síť na vytvoření nového modelu. Nový model lze vytvořit dvěma způsoby a to, vytvořením čistého nového modelu, naučeného pouze na rozpoznání aktuálně hledaných objektů nebo upravením již naučeného modelu. Pro svou bakalářskou práci jsem zvolil úpravu naučeného modelu z důvodu malé výpočetní síly použitého jednodeskového počítače. Upravený model je lépe optimalizovaný na výpočetní sílu Raspberry Pi, s tím že má nižší přesnost rozpoznání bez potřeby úpravy rychlosti analýzy. Mnou použitý upravený model se nazývá `ssd_mobilenet_v1_coco` a to konkrétně verze z roku 2017, která již byla předem ověřena ohledně použitelnosti na Raspberry Pi. Připravenou neuronovou síť s fotografiemi a zvoleným modelem je poté možné začít učit.

Samotné učení neuronové sítě probíhá využitím procesoru i grafické karty dané výpočetní jednotky. Proces probíhá po krocích, při kterých se vypisuje čas kroku a odchylka zjištěná při daném

kroku. Kroky se dle vlastního nastavení rozdělují do dávek. Méně dávek na jeden krok umožňuje rychlejší krokování procesu, nikoliv ovšem přesnější, takže proces může trvat i déle. Mnou zvolený proces krokoval po sedmi dávkách. Díky silnějšímu výpočetnímu výkonu, který jsem měl k dispozici, trval jeden krok zhruba 0,2 s. Začáteční odchylka krokování byla kolem hodnoty 8. Je doporučeno vyčkat s koncem krokování do průměrné hodnoty odchylky 10 procent oproti původní hodnoty odchylky. Hodnoty odchylky lze vykreslovat pomocí funkce neuronové sítě tensorboard. Při klesnutí odchylky pod 10 procent oproti původní hodnotě se uložil model a učení krokování se ručně zastavilo. Získaný model, použitý pro naváděcí systém, došel do 36 tisíc kroků. Na Obrázku 3.5 lze vidět postup modelu do 8 tisíc kroků.



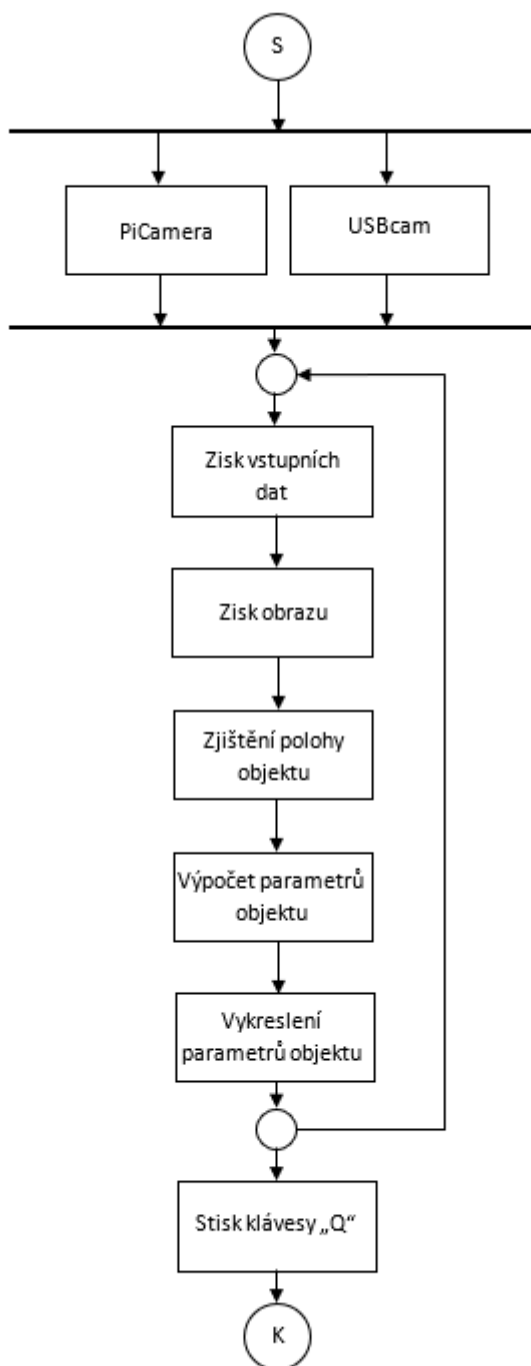
Obrázek 3.5: Graf odchylky učícího krokovacího procesu

3.3 Implementace procesu naváděcího systému

Spuštění procesu navádění předchází zvolení uživatelem, zdali chce využít kamerový modul Raspberry Pi či USB kameru. Pro zvolení kamerového modulu není nutno dělat nic, pouze spustit příslušnou aplikaci Python, jelikož je kamerový modul zvolen jako základní možnost využívaná ke snímání. Pokud je uživatel rozhodnut využít USB kameru, ke spuštění Python aplikace stačí do konzole dopsat „--usbcam“. Automaticky tak systém začne s využitím USB kamery. Proces lze vypnout kdykoliv stiskem klávesy q.

Po spuštění procesu je nutné získat vstupní data hledaných objektů, konkrétně zvolené a natrénované referenční objekty. Následně je nutné z předem vybrané kamery získat obraz, na kterém jsou referenční objekty hledány. Po zisku obrazu a vstupních dat je nutné, pomocí modelu vytvořeného neuronovou sítí, rozpoznat referenční objekt a také jeho polohu na obrazu. Poloha obrazu je vypsána a následně vykreslena do získaného obrazu. Také je zde, skrz neuronovou síť, zjištěno, s jakou přesností byl referenční objekt nalezen. Zde jsem zvolil úroveň rozpoznání, v kódu nazvanou jako threshold, která při menší hodnotě rozpoznání objekt nevykreslí a je pro něj jako nerozpoznaný. Úroveň zvolena po mnoha testech byla 70 procent, při čemž jsou všechny hodnoty nad tuto úroveň vykresleny a zároveň vypsány do obrazu. Při úspěšném nález referenčního objektu je nutné také nalézt referenční směrový bod objektu. Pokud není nalezen referenční směrový bod objektu, lze vypočítat pouze vzdálenost referenčního objektu vůči předem zvolenému cíli. Ideální stav je rozpoznání jak referenčního objektu, tak jeho směrového bodu, který zároveň leží uvnitř oblasti rozpoznaného referenčního objektu. Při takové situaci nastává další krok navádění, a to výpočet parametrů. Vypočítané parametry jsou celková vzdálenost objektu od cíle, vzdálenost na ose x a na ose y, úhel natočení vůči cíli a také rychlost snímání

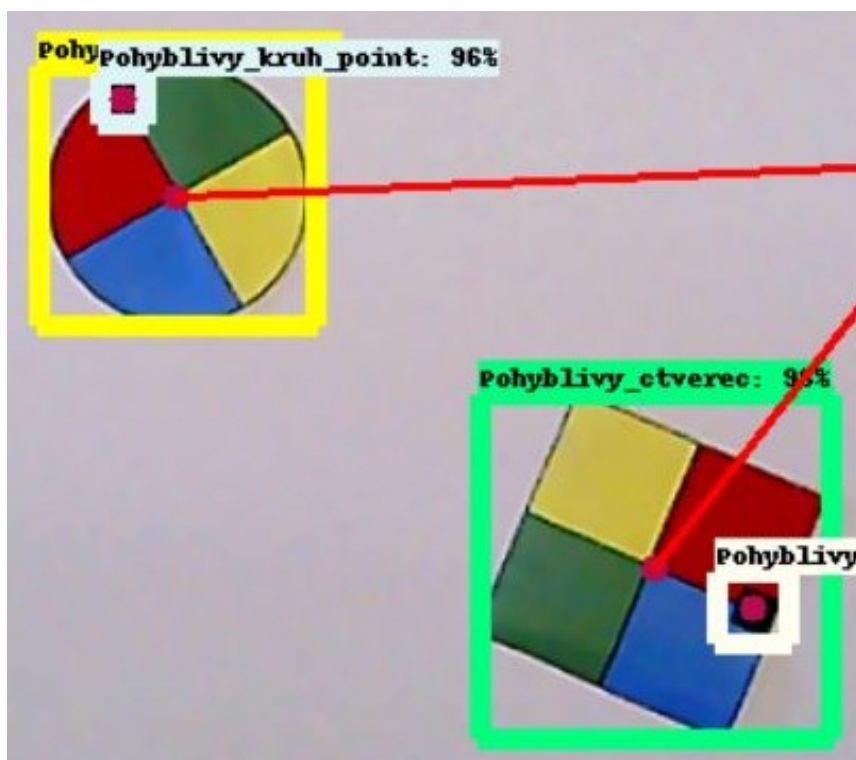
kamery. Následně se vypočítané parametry vypíší buď do obrazu či do konzole a vykreslí se směrové šipky, ukazující směr objektů do cíle. Při situaci, že objekt se nachází v cíli, začne odpočet času, konkrétně 10 snímků. Jakmile je objekt v cíli potvrzen, je vypsána informace „Jste v cíli!“ a začne 30 snímků trvající pauza. Po dobu pauzy nelze navádět objekty, aby byla možnost se přesunout pryč a začít znovu navádět. Celý proces je vykreslen vývojovým diagramem na Obrázku 3.6.



Obrázek 3.6: Vývojový diagram naváděcího systému analyzujícího obrazový signál

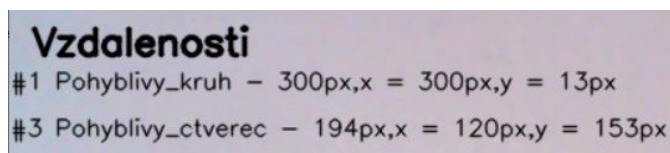
3.4 Vizualizace originálního naváděcího systému analyzujícího obrazový signál

Originální naváděcí systém analyzující obrazový signál má mnoho možností, které lze využít k následnému navádění. Systém snímá obraz v reálném čase a analyzuje jej na jednodeskovém počítači. Dokáže v obrazu rozeznat referenční objekty, určené k rozpoznání naváděného objektu v prostoru a určit a vykreslit jejich místo zobrazení. Také zvládne rozeznat u referenčních objektů jejich směrové body a také určit a vykreslit jejich místo zobrazení v prostoru.



Obrázek 3.7: Rozeznané referenční objekty a jejich směrové body

Systém dokáže také vypočítat vzdálenost daných referenčních objektů od předem zvoleného cíle. Vypočítaná vzdálenost je v px a jedná se o vzdálenost celkovou a na ose x a ose y. Tyto hodnoty jsou vypsány do obrazu. Dále jsou z těchto hodnot vypočítány hodnoty reálné vzdálenosti objektů v cm, zaokrouhleny a vypsány do konzole jednodeskového počítače. Vzdálenosti vypsané v konzoli se píší v pořadí procentuální úspěšnosti rozeznání objektů. Pro směrové body objektů nejsou vypsány vzdálenosti, takže může nastat situace jako na Obrázku 3.8, kdy je vypsán pohyblivý kruh jako nejpřesnější a pohyblivý čtverec až jako třetí nejpřesněji rozeznatý objekt.



Obrázek 3.8: Vzdálenosti vypsané do obrazu v px

```
Vzdalenost Pohyblivy_kruh k cili je 30cm
Vzdalenost Pohyblivy_ctverec k cili je 19cm
```

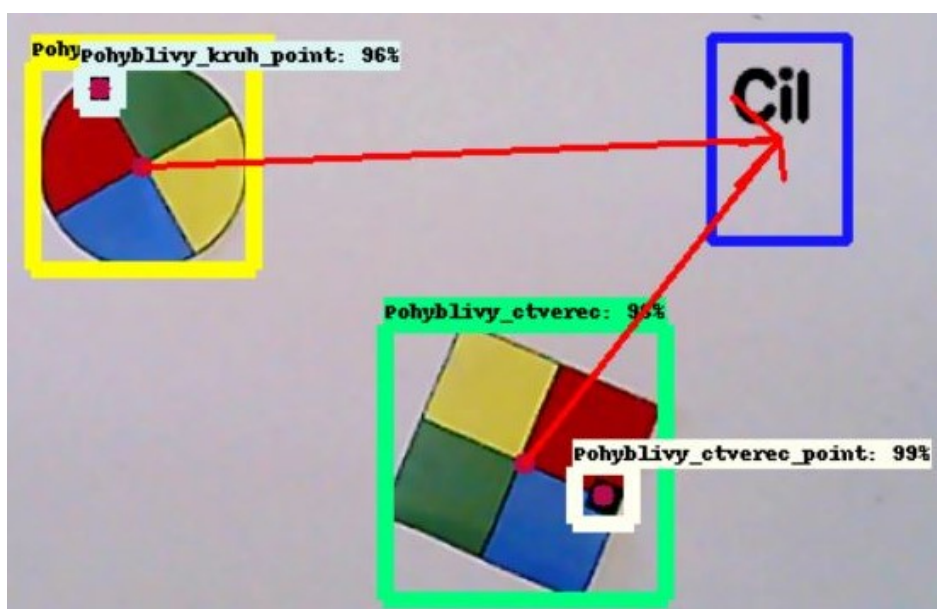
Obrázek 3.9: Vzdálenosti vypsáné do konzole v cm

Systém také na základě rozeznání směrového bodu objektu dokáže zjistit úhel natočení referenčního objektu. Úhel natočení je vypočítán z trojúhelníku tvořeného cílem, středem referenčního objektu a směrovým bodem objektu. K vypočítání je nutné rozeznat objekt i jeho směrový bod a určit, že se směrový bod daného objektu nachází uvnitř rozeznatého objektu. Úhel je následně vypsán také do konzole.

```
Uhel objektu Pohyblivy_kruh k cili je 117.08828575926222
```

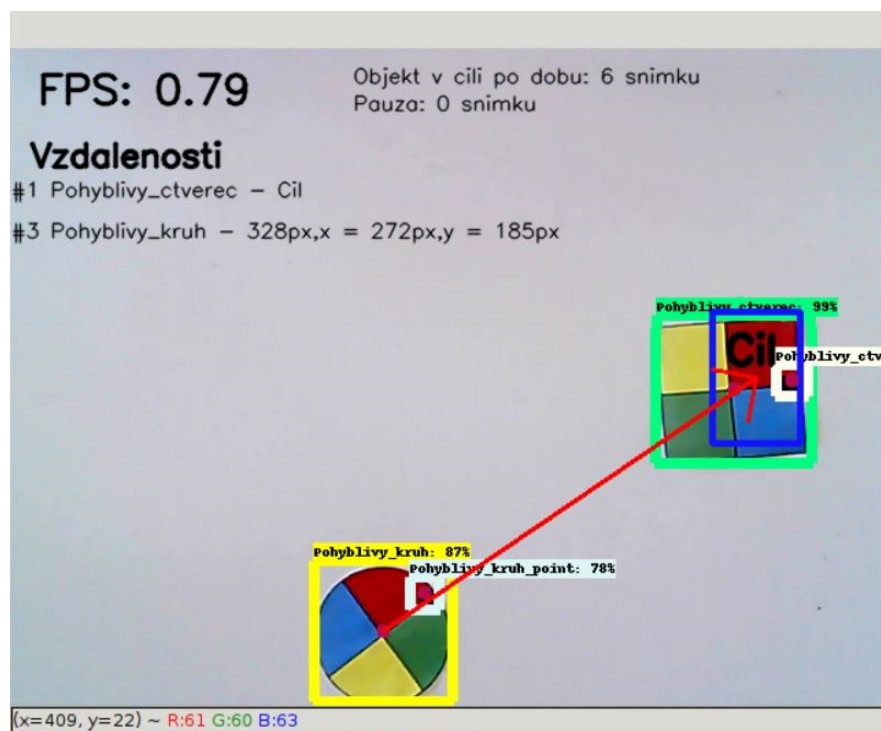
Obrázek 3.10: Úhel natočení objektu vůči cíli

Naváděcí systém ukazuje směr natočení nejen výpisem vypočítaného úhlu, ale také graficky znázorňuje směrovou šipku, která ukazuje směr, kterým se má objekt vydat směrem do cíle.



Obrázek 3.11: Směrové šipky ukazující směr k cíli

Systém rozezná, pokud se jeden z objektů dostane do cíle. Přestane počítat vzdálenost a začne počítat počet snímků, po jakou dobu zůstane objekt v oblasti cíle. Pokud těchto snímků bude přes deset, systém vypíše „Jste v cíli!“ a začne pauza. Pauza trvá třicet snímků, aby se objekty dostaly z cíle pryč. Po uplynutí se znovu začne rozpoznávat, zdali je objekt v cíli nebo ne. Kromě toho systém také ukazuje, pod pojmem „FPS“, jakou rychlostí počtu snímků za sekundu je prostor snímán a analyzován jednodeskovým počítačem.



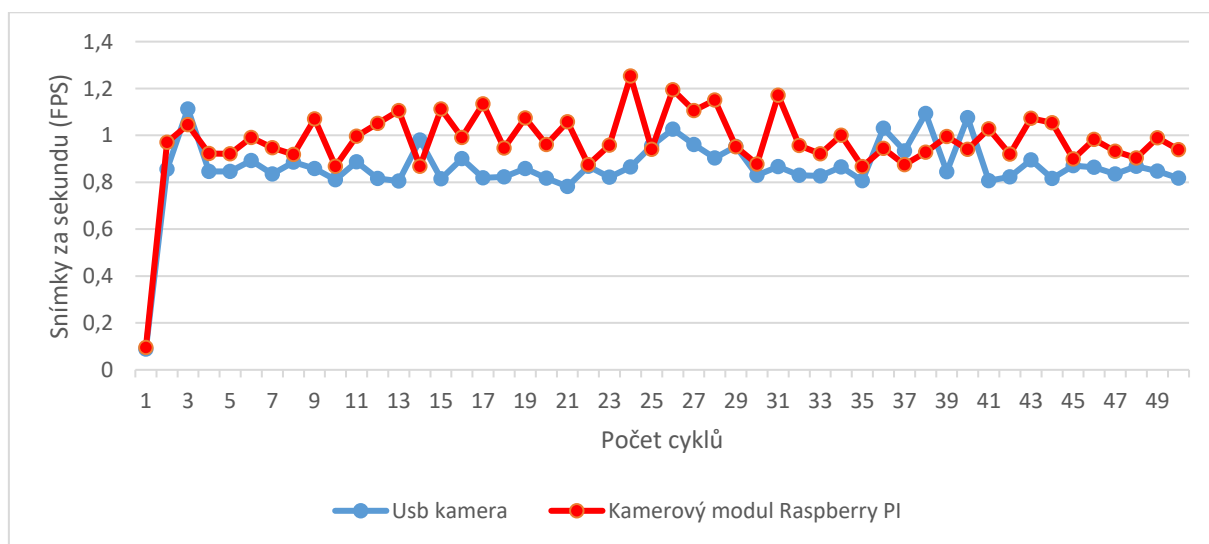
Obrázek 3.12: Objekt v cíli

4 Měření a analýza navádění

Naváděcí systém analyzující obrazový signál je potřeba prověřit v oblasti měření a zanalyzovat jeho funkčnost, přesnost a rychlost. Jsou verifikovány oblasti rychlosti snímání a analýzy jednodeskového počítače, jeho přesnosti ve vzdálenosti a úhlovém natočení a úspěšnosti rozeznání objektů. Také jsou popsány rozdíly při použití rozdílných vstupních kamer, ať už se jedná o kamerový modul Raspberry Pi či USB kameru v oblasti zpoždění snímání.

4.1 Verifikace rychlosti snímání

Rychlost snímání obrazu je dána výpočetní silou používaného hardwaru. Jednodeskový počítač Raspberry Pi 3B+ je relativně výkonný, ovšem jelikož se zde jedná o zpracování obrazu, navíc pomocí knihoven Tensorflow a OpenCV, a následně o analýzu daného zpracovaného obrazu, dostáváme se na průměrnou hodnotu okolo jednoho snímku za sekundu. Výpočet této rychlosti zajišťuje právě OpenCV knihovna a její funkce „Get tick frequency“. Vypočítaná rychlost se se zvyšujícím počtem objektů nesnižuje, respektive snížení rychlosti není možné rozeznat. Dle mých vypočítaných měření hodnota snímků za sekundu kmitá v oblasti od 0,73 do 1,22. Tyto údaje jsem verifikoval sběrem dat během snímání a následným vykreslením do Obrázku 4.1. Hodnoty vykreslené v grafu ukazují kmitání kolem hodnoty jednoho snímku za sekundu. Tedy kromě první hodnoty snímání, u které je nutno vzít v potaz inicializaci programu. Z grafu lze také vyčíst, že USB kamera měla o trochu nižší rychlost snímání, což přisuzuji jinému typu připojení a také jiné optimalizaci vůči optimalizovanému kamerovému modulu.



Obrázek 4.1: Graf porovnání rychlosti snímání obou kamerových systémů v čase

4.2 Verifikace přesnosti snímání vzdáleností

Verifikace přesnosti snímání vzdáleností lze udělat přesným měřením reálné vzdálenosti pomocí pásma a porovnání s hodnotami naměřenými v aplikaci. Vzdálenosti zobrazené v obrazu jsou v

pixelch, zatímco reálně jsme schopni naměřit vzdálenost na milimetry. Převedení pixelů na milimetry je provedeno za porovnání velikosti snímané plochy vůči ploše naměřené. Snímaná plocha kamerou má rozměry 640 x 480 pixelů a naměřená snímaná oblast má rozměry 640 x 480 mm. Poměr je tedy 1:1. V tabulce 4.1 jsou vypsané porovnané hodnoty. Jak lze vidět, výsledky vypočítaných vzdáleností objektů od cíle s těmi naměřenými mají maximální odchylku 2 mm. Nepřesnost vypočítané hodnoty je dána naučeným modelem, který byl naučen na referenční objekty s lehkou nepřesností kvůli jednoduššímu rozpoznání. Naměřených i vypočítaných hodnot je patnáct.

Tabulka 4.1: Porovnání vzdáleností

Vypočítaná vzdálenost [mm]	Naměřená vzdálenost [mm]	Odchylka [mm]
260	260	0
343	342	1
350	348	2
266	266	0
408	407	1
485	485	0
148	147	1
492	490	2
416	414	2
195	196	1
234	232	2
218	216	2
318	319	1
246	246	0
191	192	1

4.3 Verifikace přesnosti snímání úhlů

Verifikace přesnosti snímání úhlů je způsob ověření správnosti vypočítaných úhlů aplikací vytvořenou pro navádění s úhly naměřenými. Naměřit úhel, který byl v aplikaci vypočítán, se podařilo pomocí úhlového pravítka. Převedení hodnot není třeba, jelikož jak vypočítané, tak měřené hodnoty jsou ve stupních. Odchylka je zde větší, pohybuje se až okolo hodnoty 5°. Tato odchylka je dána více vstupními informacemi potřebnými k změření úhlu. Je zde nutné počítat s nepřesností rozpoznání objektu, stejně jako u vzdálenosti. Dále je nutné zjistit, zdali byl dobře rozpoznán také směrový bod objektu a tyto dva body propojit. Jakákoliv nepřesnost zde potom ovlivňuje odchylku vypočítaného úhlu vůči naměřenému. Naměřených i vypočítaných hodnot je patnáct.

Tabulka 4.2: Porovnání úhlů

Vypočítaný úhel [°]	Naměřený úhel [°]	Odchylka [°]
96	94	2
172	171	1
99	98	1
9	14	5
30	30	0
68	73	5
83	82	1
74	79	5
159	157	2
30	31	1
103	100	3
78	77	1
121	123	2
148	151	3
54	54	0

4.4 Verifikace rozpoznání referenčních objektů

Rozpoznání referenčních objektů je aplikováno za pomoci neuronové sítě Tensorflow, která společně s knihovnou OpenCV zpracovává a analyzuje obraz z kamery. Přesnost rozpoznání objektů je stejně tak vnitřní funkcí, jejíž chod nelze narušovat. Výstupní složkou rozpoznání objektů je tedy pouze procentuální hodnota, na kterou neuronová síť daný objekt rozpoznala. Mnou zvolena je úroveň, nad kterou již program hodnotí objekt jako rozeznatý. Dle získaných zkušeností jsem tuto hodnotu, v kódu nazvanou jako Threshold, dal na 70 procent. Pokud je hodnota rozpoznání menší, není objekt vypsán ani vykreslen jako rozpoznatý.

4.5 Analýza rozdílů použitých kamerových zařízení

Použití kamerových zařízení je nutné kvůli snímání obrazu. Mnou vybrané kamerové zařízení, tedy kamerový modul Raspberry Pi a USB kamera A4Tech PK-910H, byly rozlišné, nicméně jejich rychlost snímání byla stejná. Průběh jejich rychlostí snímání lze vidět na přiloženém Obrázku 4.1. Obě kamerové zařízení se také nelišily ve svém rozlišení s tím, že obě disponovaly HD rozlišením 1080p30, což znamená že rozlišení bylo 1920x1080 px a zařízení snímala maximální rychlosti 30 fps. V čem se použité kamerové zařízení lišily bylo jejich připojení, kdy kamerový modul disponuje FLEX kabelem, který je použit přímo pro CSI rozhraní Raspberry Pi, zatímco USB kamera byla připojena díky rozhraní USB 2.0. Rozdíl připojení je také zřejmě důvodem zpoždění, kterou disponuje USB kamera. Toto zpoždění není detekováno v případě kamerového modulu. Zde může být důvod, že kamerový modul, na rozdíl od USB kamery, je již optimalizován na práci s Raspberry Pi a jeho malá výkonnost nezvládá neoptimalizovaný software USB kamery rychle zpracovat. Z tohoto důvodu je nakonec reálně použit kamerový modul Raspberry Pi.

5 Zhodnocení dosažených výsledků

Zhodnocení dosažených výsledků mé bakalářské práce je náročný úkol, jelikož si subjektivně myslím, že se práce vydařila a je funkční. Nicméně bych ji rád ještě zdokonalil, pokud by k tomu byl čas. Chtěl bych nejprve shrnout, co se mi podařilo, a nakonec také zmínit, kde cítím ve své bakalářské práci jisté rezervy. Osobně musím říct, že největším výsledkem a zároveň přínosem pro mne byla práce s neuronovou sítí Tensorflow. Výsledkem můžu označit natrénovaný model naučený rozpoznávat nově určené referenční objekty. Jako přínos vidím veškerou práci s danou sítí a také získání povědomí o tom, co to vůbec neuronové sítě jsou. Také jsem zjistil mnohé o strojovém učení a celkovém zpracování obrazu. Velice mě u neuronových sítí a strojovém učení těší, že se jedná o poměrně novou oblast programování a práce v ní ještě není několikrát omílaná. Tato oblast se může posunout dále a já bych se v budoucnu tímto směrem chtěl také profilovat.

Řekl bych, že v oblasti navádění jsou výsledkem zjištěné a vypočítané hodnoty, které k navádění potřebujeme. Rozeznání referenčního objektu a tím i zjištění polohy objektu je základním kamenem ke zjištění dalších informací potřebných k navádění. Objekt, který je označen referenčním objektem, může dostávat informace nejen o své poloze, ale také o poloze cíle, jak daleko cíl od něj leží, jak je daleko na ose x a ose y a také pod jakým úhlem vůči objektu lze cíl najít. Je zajištěno, že pokud objekt dorazí do cíle, po chvíli kontroly správného snímání dojde k přenosu informace, že se objekt nachází v cíli. Pokud se objekt v cíli nachází, nedochází k žádnému zisku dat, takže se objekt dále nepohybuje. Po zjištění existence objektu v cíli může objekt cíl opustit a po dané časové hodnotě se spustí analýza znovu.

Všechny informace jsou zjištěny za pomoci kamery, buď USB či kamerového modulu Raspberry Pi, a analyzovány jednodeskovým počítačem Raspberry Pi. Pro mne je také podstatné, že jsem se naučil, jak s jednodeskovým počítačem Raspberry Pi 3B+, tak také s jeho příslušenstvím, jako je kamerový modul či USB kamera. Jednodeskový počítač běží na verzi Linuxu Raspbian Stretch, který se mi podařilo naučit a následně v něm pracovat. Je nutno podotknout, že jsem již měl předchozí zkušenosti s prací s Raspberry Pi. Program jako takový běží v programovacím jazyku Python. Ten pro mne byl novinkou, nicméně jsem získal mnoho zkušeností a informací, které bych bez této bakalářské práce nenabyl.

Zdokonalení práce bych viděl v optimalizaci či za použití jiného hardwaru. Použitý hardware totiž výkonnostně zvládl vykonat jak rozeznání objektu, tak následnou analýzu. Problém je rychlost snímání, kolem jednoho snímku za sekundu, která není ideální. Ovšem neodporuje to použití, takže to vidím pouze jako možnost zdokonalení. V oblasti dat, je možnost data nejen zjišťovat a vypočítat, ale také posílat. Vzhledem k tomu, že není zatím vytvořený objekt, který by dané navádění zužitkoval, žádnou možnost přenosu dat jsem nevytvářel. Tato možnost by byla ideálně vedena přenosem přes síť Wi-Fi, jelikož Raspberry Pi 3 je první model, který má tuto možnost v základní konfiguraci a není nutné dokupovat moduly či připojovat ethernet kabelem k síti.

Závěr

Cílem bakalářské práce bylo vytvořit naváděcí systém analyzující obrazový signál. Základní částí bylo teoreticky zjistit, jakým způsobem můžeme obrazový signál analyzovat a také jak vytvořit naváděcí systém. Po zjištění teoretických informací bylo nutné vytvořit návrh naváděcího systému.

Návrh naváděcího systému analyzujícího obrazový signál se skládal z dvou hlavních částí. Bylo nutné navrhnout hardwarovou část systému a s tím také dále pracovat i v souvislosti s částí softwarovou. Mezi softwarovou část patří jistě zvolený operační systém našeho hardwaru a také zvolený programovací jazyk, který byl na systém bakalářské práce použit. Můžeme zde také počítat rozbor problematiky navádění a určení referenčních objektů, použitých k rozeznávání naváděného objektu. V návrhu je také nutné představit umělou neuronovou síť a strojové učení, které sloužilo k následné realizaci naváděcího systému.

Realizace naváděcího systému analyzujícího obrazový signál byla částí, kde se zjistilo, jaký hardware budeme konkrétně používat a došlo zde na celkovou realizaci testovacího přípravku. Implementoval se zde učící proces a také se zde implementoval naváděcí systém. V realizaci byl následně vizualizován celý naváděcí systém a jeho jednotlivé části, které počítá, zjišťuje a následně vykresluje. Mezi tyto vyobrazené části patří rozpoznání referenční objekt, nalezení jeho lokace, určení cíle, výpočet jeho vzdálenosti od cíle, rozeznání směrového bodu objektu, výpočet úhlu natočení objektu vůči cíli a zobrazení, pokud se objekt dostane do cíle.

Následně byla provedena verifikace celého systému, na základě vypočtených hodnot systémem a následně naměřených kontrolních hodnot. Verifikace probíhala v oblasti rychlosti snímání obrazu dvěma různými kamerovými systémy, dále porovnání přesnosti vypočtení vzdálenosti objektu od cíle. Jednalo se jak o celkovou vzdálenost, tak o vzdálenost na ose x a ose y. Také porovnání přesnosti vypočtení úhlového natočení objektu vůči cíli, a nakonec verifikace rozpoznání objektu. Výsledky verifikace byly úspěšné, kdy maximální odchylka celkové vzdálenosti objektu od cíle byla 0,2 cm a maximální odchylka úhlu natočení vůči cíli byla 5°. V měření a analýze navádění bylo ještě zmíněno, že byl rozdíl mezi oběma použitými kamerovými systémy.

Nakonec byla celé práce z pohledu autora zhodnocena. Nebylo popsáno jen to, co se povedlo a v čem byla bakalářská práce naučná, ale také to, v čem práce může být ještě zdokonalena do budoucna.

Použitá literatura

- [1] SOJKA, Eduard. *Digitální zpracování a analýza obrazů*. Ostrava: VŠB-Technická univerzita, 2000. ISBN 80-707-8746-5.
- [2] SOBOTKA, Zdeněk a Martin SOBOTKA. *Počítačová analýza a rozpoznávání obrazu*. Praha: Dům techniky ČSVTS, 1990. ISBN 80-020-0739-5.
- [3] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. BEN-Technická literatura, 2008. ISBN 978-80-7300-233.
- [4] LIČEV, Lačezar. *Analýza, modelování, rozpoznávání a vizualizace procesu měření objektů na snímcích*. Brno: Computer Press, 2010. ISBN 978-80-251-3296-8.
- [5] SMITH, Steven W. *The scientist and engineer's guide to digital signal processing*. San Diego, Calif.: California Technical Pub., c1997. ISBN 09-660-1763-3.
- [6] STRAKA, Stanislav. *Segmentace obrazu*. Brno, 2009. Diplomová práce. Masarykova univerzita. Vedoucí práce Mgr. Radka Pospíšilová.
- [7] Electromagnetic spectrum. *Wikipedia* [online]. [cit. 2019-04-29]. Dostupné z: https://en.wikipedia.org/wiki/Electromagnetic_spectrum
- [8] CMYK barevný model. *WikiSkripta* [online]. 2008, 18. 6. 2018 [cit. 2019-04-29]. Dostupné z: https://www.wikiskripta.eu/w/CMYK_barevn%C3%BD_model
- [9] PAVELEK, M.; JANOTKOVÁ, E.; ŠTĚTINA, J. *Vizualizační a optické měřicí metody*. URL <http://dt.fme.vutbr.cz/~pavelek/optika/>: FSI VUT. Brno, 2001.
- [10] VOLNÁ, Eva. *Neuronové sítě 1: Studijní materiály pro distanční kurz* [online]. Ostrava, 2008 [cit. 2019-04-29]. Dostupné z: http://www1.osu.cz/~volna/Neuronove_site_skripta.pdf. Skripta. Ostravská univerzita.
- [11] VLČEK, Michael. *Strojové učení ve strategických hrách* [online]. Brno, 2018 [cit. 2019-04-29]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor verejne.php?file_id=180937 . Semestrální projekt. Vysoké učení technické v Brně. Vedoucí práce Doc. RNDr. PAVEL SMRŽ, Ph.D.
- [12] Neuronové sítě. *Mendelova univerzita v Brně* [online]. Brno [cit. 2019-04-29]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471
- [13] Template Matching. *OpenCV documentation* [online]. [cit. 2019-04-29]. Dostupné z: https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html

Seznam příloh

Součástí bakalářské práce je DVD.

Seznam příloh na DVD

V adresáři „Zdrojový kód“ nalezneme zdrojový kód aplikace pod názvem Pohyblivy_objekt.py, naučený model neuronové sítě na rozeznávání referenčních objektů nazvaný inference_graph_point_36k a soubor labelmap.pbtxt určující podstatné informace pro neuronovou síť.

V adresáři „Neuronová síť“ nalezneme podstatné součásti na učení modelu neuronové sítě, včetně složky s předchozím modelem, složky images s obrázky potřebnými k naučení sítě a csv soubory.

V souboru „InformaceKeSpusteni“ nalezneme informace k spuštění jak programu, tak celého učení sítě.

V souboru „Video_BP“ nalezneme předtočené video fungujícího naváděcího systému.

V souboru „PorovnaniSystemu“ nalezneme informace o změřených hodnotách fps obou kamerových systémů.
